Studuino プログラミング環境

取扱説明書

本資料は、Studuino(スタディーノ)プログラミング環境の取扱説明書になります。
Studuino プログラミング環境の変更に伴い、加筆・修正が加えられる可能性があります。また、本取扱説明書は、下記の資料を参照します。

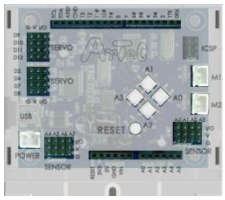
■Studuino ソフトウェアのインストール
URL:http://artec-kk.co.jp/studuino/docs/jp/Studuino_setup_software.pdf

目次

1. Robotist と Studuino プログラミング環境について 1
   1.1. 概要・特徴 1
   1.2. プログラム動作環境 2
   1.3. Studuino について 2
2. 導入 5
   2.1. デバイスドライバのインストール 5
   2.2. プログラミング環境のインストール・アンインストール 5
3. スタートアップ画面 6
4. Studuino (スタディーノ) アイコンプログラミング環境 8
   4.1. 概要・特徴 8
   4.2. アイコンパレット 9
   4.3. プログラムエリア 11
   4.4. 属性エリア 17
   4.5. メインメニュー 22
   4.6. コンテキストメニュー 32
5. Studuino (スタディーノ) ブロックプログラミング環境 33
   5.1. 概要・特徴 33
   5.2. コマンドグループパレットとブロックパレット 34
      5.2.1. ブロックについて 34
   5.3. スクリプトエリア 45
   5.4. コンディションエリア 45
   5.5. メインメニュー 45
   5.6. コンテキストメニュー 57
6. トラブルシューティング 59
   6.1. Studuino プログラミング環境共通 59
   6.2. Studuino アイコンプログラミング環境 60
   6.3. Studuino ブロックプログラミング環境 61
7. お問い合わせ先 64

1. Robotist と Studuino プログラミング環境について

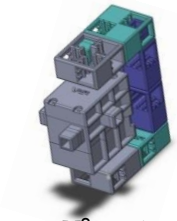
1.1. 概要・特徴



Arduino 互換基板

Studuino

アーテック
ブロック



パーツ

[モーター、LED
ブザー、センサー]

Studuino
プログラミング環境

ブロックとパーツを
組み合わせて
ロボットの外観を作成

初級者～上級者に合わせた
プログラミング環境で
ロボット制御プログラムを作成

Robotist(ロボティスト)は、Studuino(スタディーノ)、アーテックブロック、パーツ、Studuino プログラミング環境から成ります。ユーザーは、Studuino とアーテックブロックとパーツを組み合わせ、自由にロボットの外観と構造を作成し、Studuino プログラミング環境を利用して、ロボットを自由に動作させることができます。

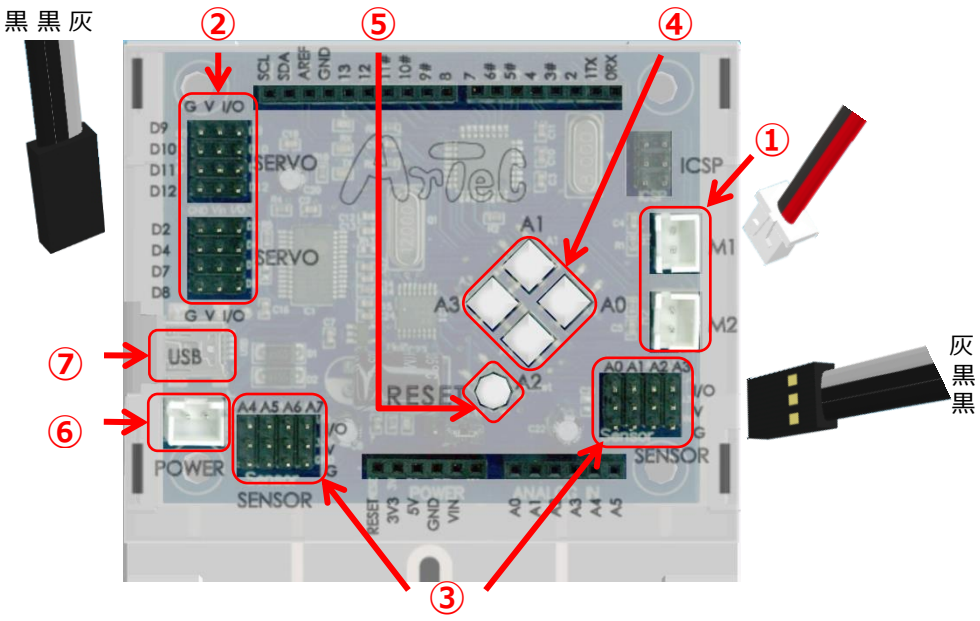
Studuino プログラミング環境は、ドラッグ＆ドロップによる操作とプログラム作成をサポートする機能により、プログラミング初心者でも本格的なロボット制御プログラムを作成できる環境となっています。また、Arduino 言語への翻訳もサポートしているため、Studuino プログラミング環境で作成したプログラムを Arduino 言語に翻訳し、より高度なロボット制御プログラムを作成することもできます。

1.2. プログラム動作環境

仕様	内容
OS	Windows XP(SP3 以上適用済み) / Windows Vista / Windows7 / Windows8 / Windows8.1
プロセッサ(CPU)	Pentium4 2GHz 以上または相当品推奨
メモリ	256MByte 以上
USB	USB2.0 対応ポート
ソフトウェア	Microsoft .NET Framework4.0 が必要 (Studuino プログラミング環境インストール時に自動的にインストールされます)
画面サイズ	XGA(1024×768)以上

1.3. Studuino について

ロボット作成時に使用する、Studuino の入出力部について説明します。



① DC モーター用コネクター

DC モーターを接続するコネクターです。DC モーターは 2 つまで接続可能で、コネクター名は、M1, M2 です。

② サーボモーター用コネクター（デジタルピン用コネクター）

サーボモーターを接続するコネクターです。サーボモーターは 8 つまで接続可能で、コネクター名は、D2, D4, D7, D8, D9, D10, D11, D12 です。サーボモーターを接続

する際は、信号線(灰色の線)が内側になるように接続して下さい。

③ センサー/LED/ブザー用コネクター（アナログピン用コネクター）
センサー、LED、ブザーのいずれかを接続するコネクターです。コネクター名は、A0～A7 です。タッチセンサー、加速度センサー以外のセンサーは、A0～A7 に接続できます。タッチセンサー、LED、ブザーが接続可能なコネクターは、A0～A5 になります。加速度センサーは、A4 と A5 にのみ接続できます。センサー/LED/ブザーを接続する際は、信号線(灰色の線)を内側になるように接続して下さい。

④ プッシュスイッチ
ボタンスイッチです。スイッチ名は、A0～A3 です。プッシュスイッチを使用する場合は、センサー/LED/ブザー接続コネクターの A0～A3 は使用できません。

⑤ リセットスイッチ
プログラムをリセットするスイッチです。Studuino を初期化した後や、PC が Studuino を正しく認識できないとき、その他 Studuino の動作がおかしいときなどには、このスイッチを押して Studuino を再起動してください。

⑥ 電源コネクター（外部電源用コネクター）
電源を接続するコネクターです。センサー/LED/ブザー/スイッチは、USB からの供給電源で動作しますが、DC モーターやサーボモーターを動かす場合は電源コネクターから電力を供給する必要があります。また、USB ケーブルを外し、Studuino を動作させるときにも電源コネクターから電力を供給する必要があります。

⑦ 通信ケーブル接続コネクター（USB mini B コネクター）
PC との通信を行うために通信ケーブルを接続するコネクターです。コネクターの形状は USB の mini-B 規格であり、製品に付属の通信ケーブルに限らず、市販の USB ケーブルを接続することができます。

また、下記のコネクターまたはスイッチは同時に使えません。

- **DC モーター接続コネクターM1 とサーボモーター用ネクターの D2, D4**
- **DC モーター接続コネクターM2 とサーボモーター用コネクターの D7, D8**
- **プッシュスイッチ A0 とセンサー/LED/ブザー用コネクターA0**
- **プッシュスイッチ A1 とセンサー/LED/ブザー用コネクターA1**
- **プッシュスイッチ A2 とセンサー/LED/ブザー用コネクターA2**
- **プッシュスイッチ A3 とセンサー/LED/ブザー用コネクターA3**

たとえば、M1 に DC モーターを接続し、使用する場合、D2, D4 にサーボモーターを接続し

てもサーボモーターの動作保証はされません。また逆に、D2, D4 をサーボモーターとして使用する場合、M1 に DC モーターを接続しても DC モーターの動作保証はされません。

2. 導入

Studuino プログラミング環境を使用するには、デバイスドライバとプログラミング環境のインストールが必要になります。

2.1. デバイスドライバのインストール

PC から Studuino を制御する際に、デバイスドライバを PC にインストールしておく必要があります。デバイスドライバのインストールについては、別紙「**Studuino 基板 セットアップ USB デバイスドライバのインストール**」を参照してください。

2.2. プログラミング環境のインストール・アンインストール

Studuino プログラミング環境は、Windows、Mac OS X、Raspberry Pi に対応しています。Studuino プログラミング環境のインストール・アンインストールは、別紙「**Studuino プログラミング環境 Studuino ソフトウェアのインストール**」を参照してください。

3. スタートアップ画面

Windows 環境のみスタートアップ画面によりプログラミング環境を選択することができます。Mac OS X、Raspberry Pi でのプログラミング環境の起動方法は、別紙 **Studuino ソフトウェアのインストール**を参照してください。

Windows 環境の Studuino プログラミング環境は、上図のスタートアップ画面から起動します。以下にスタートアップ画面について記します。

■ アイコンプログラミング環境

アイコンプログラミング環境を選択すると、ソフトウェアが起動し、ロボット制御プログラムを作成することができます。アイコンプログラミング環境を使用したロボット制御プログラムの作成については、**4. Studuino (スタディーノ) アイコンプログラミング環境**を参照して下さい。

■ ブロックプログラミング環境

ブロックプログラミング環境を選択すると、ソフトウェアが起動し、ロボット制御プログラムを作成することができます。ブロックプログラミング環境を使用したロボット制御プログラムの作成については、**5. Studuino (スタディーノ) ブロックプログラミング環境**を参照して下さい。

■　言語設定

メインメニューの「言語設定」を選択すると、表示されている言語の設定を行うことができます。言語は日本語、英語、中国語から選択できます。

■　作例

アーテックブロック、Studuino、パーツを使用したロボットの作例とその作り方が表示されます。

■　ソフトウェアマニュアル

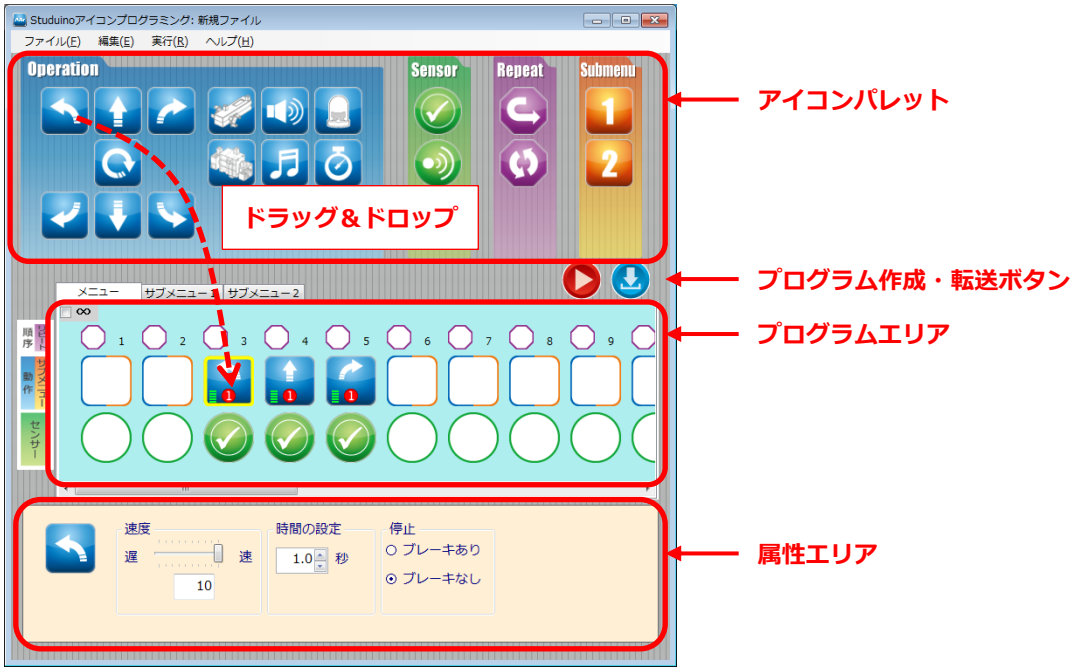
Studuino プログラミング環境のマニュアルが表示されます。

■　終了

メインメニューの「終了」を選択すると、スタートアップ画面を終了します。
各メニュー項目からメインメニューへ戻る場合、画面の左下にある「メインメニューへ戻る」ボタンをクリックします。

4. Studuino (スタディーノ) アイコンプログラミング環境

4.1. 概要・特徴

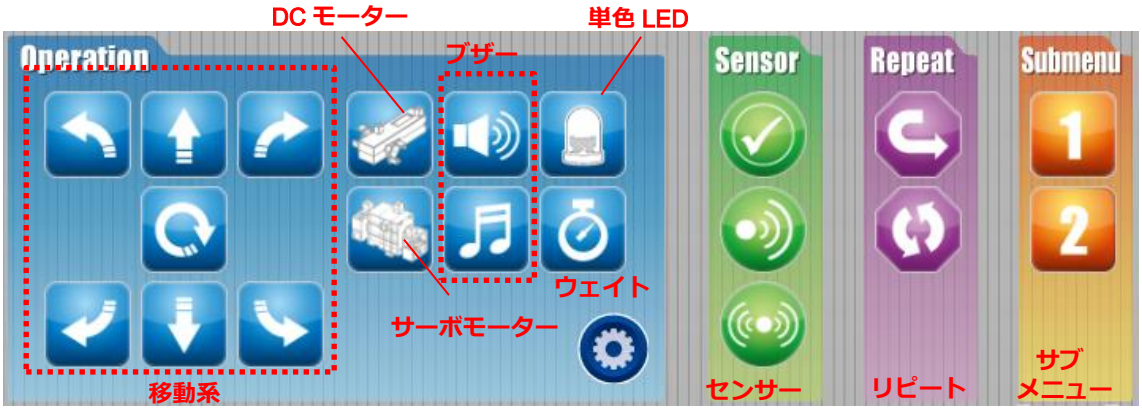


Studuino アイコンプログラミング環境は、ロボットを制御するアイコンをプログラムエリアにドラッグ＆ドロップして、ロボット制御プログラムを作成するプログラミング環境です。

プログラムエリアに置かれた各アイコンは、属性エリアで、詳細な動作の設定が可能になります。また、リアルタイムにロボットの動作を確認できるテストモード(**4.5** メインメニューのテストモードを参照)を利用したり、センサービューア(**4.5** メインメニューのセンサー値確認モードを参照)によりセンサー入力の値を確認しながら、簡単に本格的なロボット制御プログラムを作成することができます。

プログラムが完成したら、プログラム転送ボタンでプログラムを Studuino へ転送し実行することができます。また、作成した制御プログラムを Arduino(アルドゥイーノ)言語に変換し、Arduino IDE で編集することもできます(**4.5** メインメニューの Arduino 言語を表示を参照)。

4.2. アイコンパレット



アイコンパレットでは、ロボット制御アイコンを提供します。Operation グループではロボットの動作に関するアイコンを、Sensor グループではセンサー値を使用するアイコンを、Repeat グループでは制御の一部を繰り返すアイコンを、Submenu グループでは、サブメニューを実行するアイコンを提供します。

以下に各グループで提供されるアイコンを記します。

■ Operation グループ

Operation グループでは、ロボットのパーツを制御するアイコンを提供します。

DC モーターを制御

サーボモーターを制御

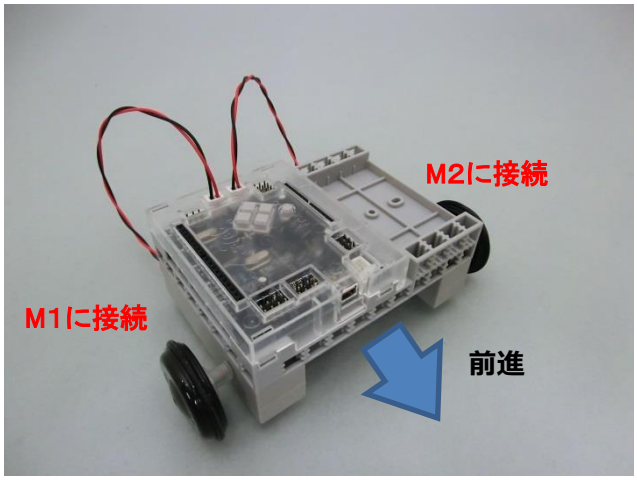
単色 LED を制御

ブザーからのメロディ出力

ブザーからの単音出力

指定時間、ウェイトする

また、下図のように DC モーター 2 つを車のタイヤとして M1, M2 に接続した場合に、車の移動制御に使用するアイコンも提供します。タイヤ側が前になります。



前進

左折(前)

右折(前)

回転

後退

左折(後)

右折(後)

■ Sensor グループ

Sensor グループでは、ロボットの動作の条件に、センサーからの入力値を使用するアイコンを提供します。

センサー入力値をロボットの動作条件に使用しない

1 つのセンサーの値をロボットの動作条件に使用する

2 つのセンサーの値をロボットの動作条件に使用する

■ Repeat グループ

Repeat グループでは、処理の一部分をリピートする際に使用するアイコンを提供します。

リピート開始点

リピート終了点

■ Submenu グループ

Submenu グループでは、サブメニューで定義した処理を実行するアイコンを提供します。

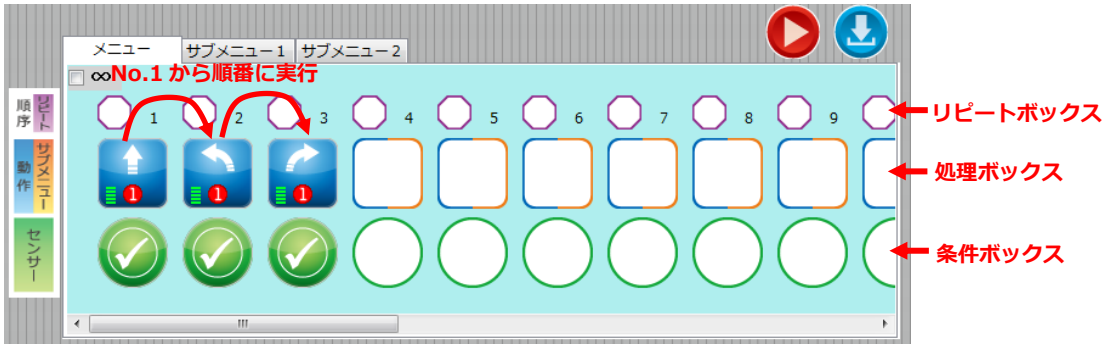
サブメニュー1 を実行

サブメニュー2 を実行

4.3. プログラムエリア

プログラムエリアは、アイコンパレットの制御アイコンをドラッグ＆ドロップで配置し、ロボット制御プログラムを作成する領域です。1 列 1 つの処理として表します。メニュータブによって、「メニュー」、「サブメニュー1」、「サブメニュー2」を切り替えることができます。

■　メニュー

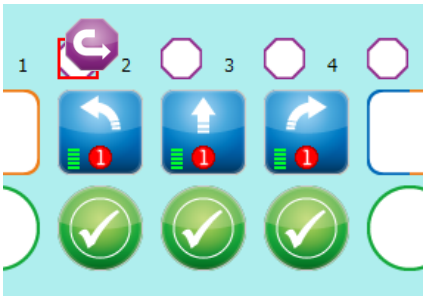


メニューでは、ロボット制御の開始から終了までをプログラムします。No.1～No.24 の 24 個の処理を設定することができます。作成したプログラムは、No.1 から順番に実行されます。メニューは、リピートボックス、処理ボックス、条件ボックスを持っており、リピートボックスに Repeat グループのアイコンを配置、処理ボックスに Operation グループまたは、Submenu グループのアイコンを配置、条件ボックスに Sensor グループのアイコンを配置することができます。

- **リピートボックス**

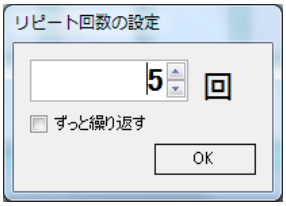
リピートボックスには、Repeat グループのアイコンをドロップすることができます。以下にリピートボックスの設定手順を記します。

① リピートしたい処理の先頭のリピートボックスにリピートアイコン(開始点)をドロップします。



② リピートしたい処理の終端のリピートボックスにリピートアイコン(終了点)をドロップします。この時、リピートされる処理が四角で囲まれます。

③ ②のドロップ後、「リピート回数の設定」ダイアログボックスが表示されるので、リピートしたい回数を設定します。また、ずっと繰り返すにチェックを入れることで、無限ループに設定できます。



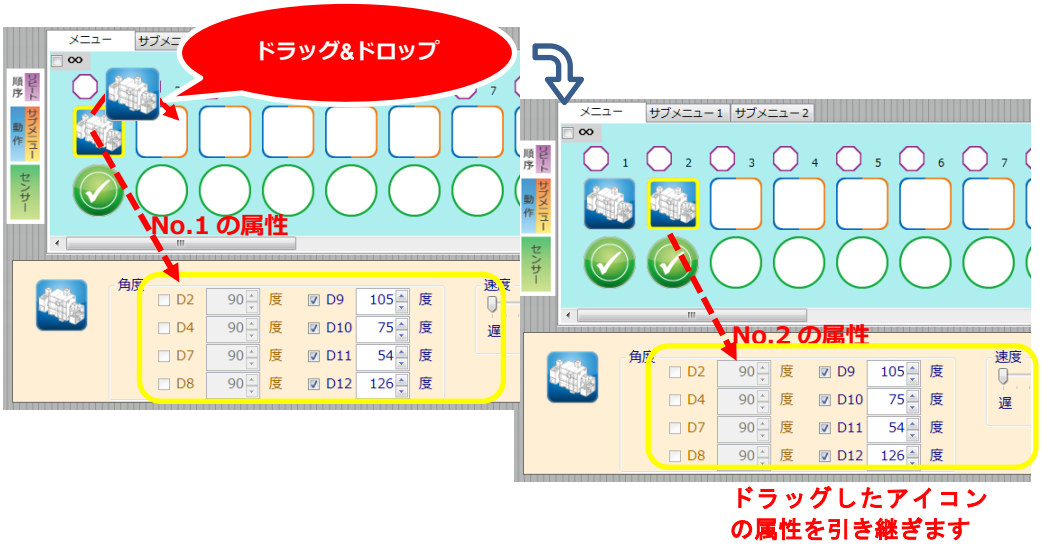
④ リピート処理が完成しました。
リピート回数の再設定は、ドロップしたリピートアイコン(終了点)をクリックし、「リピート回数の設定」ダイアログボックスを再表示することで行うことができます。

- **処理ボックス**

処理ボックスには、Operation グループまたは、Submenu グループのアイコンをドロップすることができます。サブメニュー1 アイコン1または、サブメニュー2 アイコン2は、プログラムエリアのサブメニュー1 または、サブメニュー2 で作成された制御を実行します(サブメニューの詳細は、後述のサブメニュー1,2 を参照して下さい)。

また、処理ボックス上のアイコンを別の処理ボックスにドラッグ&ドロップすることで、コ

ピーすることが可能です。下図に示すように、処理ボックスに置いたアイコンをドラッグ＆ドロップした場合、新たに作成されるアイコンは、ドラッグしたアイコンの属性を引き継ぎます。



- **条件ボックス**

条件ボックスには、同じ列の処理ボックスに置かれたアイコンを実行する条件を設定します。処理ボックスにアイコンをドロップすると自動的に無条件アイコン　が条件ボックスに表示されます。条件ボックスに、Sensor グループのアイコンをドロップすることで、センサー入力値を処理の条件に使用する事ができます。条件に使用するセンサーやしきい値は、属性エリアで設定することができます。条件ボックスに配置したセンサーアイコンを

選択することで、条件を属性エリアに表示し設定することができます。センサーアイコン

による条件の設定の詳細は、**0**

属性エリアを参照して下さい。

上の図の場合、No.1 では、無条件で車の直進処理を実行しますが、No.2 では、光センサーの値がしきい値(3)よりも小さい場合に車の後退処理を実行します。光センサーの値がしきい値(3)よりも大きい場合は、後退処理は実行しません。

■　サブメニュー1, 2

サブメニューでは、特定の小規模のプログラムを作成でき、No.1～No.8 の 8 個の処理を設定することができます。サブメニューで作成したプログラムは、メニューでサブメニューアイコンが実行されると、No.1 から順番に実行されます。サブメニューは、処理ボックス、条件ボックスを持っており、処理ボックスに Operation グループのアイコンを配置、条件ボックスに Sensor グループのアイコンを配置することができます。

■　複数アイコンの選択とドラッグ＆ドロップ

プログラムエリアに配置された複数のアイコンを選択し、ドラッグ＆ドロップによってコピーすることが出来ます。

プログラムエリアの行動・条件・リピートボックス以外の部分でマウスの左ボタンを押すと選択開始状態になり、マウスカーソルが に変わります。そのままカーソルをドラッグしていくと、選択範囲に入ったアイコンが点線で囲まれていきます。マウスの左ボタンを離すと選択状態が確定し、選択枠が太い実践に変わりマウスカーソルが元に戻ります。

※行動ブロックのみを選択することも可能です。また、空白ブロックやサブメニューアイコンが設定されているブロックは、選択範囲に含めることが出来ません。

選択範囲内のいずれかのアイコンでドラッグを開始します。このとき、マウスカーソルは一番先頭(一番左)のアイコン画像に変わります。

他の行動ブロックへドロップするとコピーが完了します。別メニューのタブへカーソルを持っていくことで、タブを切り替えることが出来ます。切り替わったメニュー内の行動ブロックにドロップするとコピーが完了します。

※行動アイコンのみをドラッグ＆ドロップした場合、コピー先の行動ブロックの状態で条件アイコンが決まります。空白（未設定）の場合、条件ブロックには無条件アイコンが設定され、設定済みの場合、条件アイコンは変更されません。

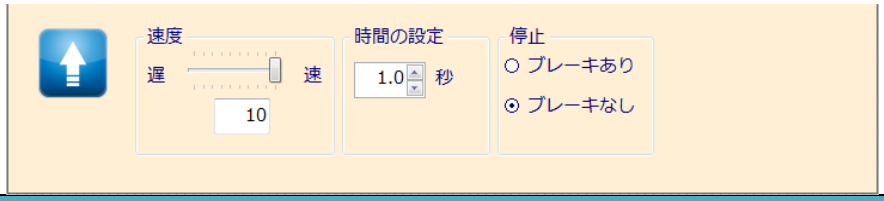
メニュー
サブメニュー1
サブメニュー2
順序
動作
センサー
行動ブロックへドラッグ＆ドロップ
別メニューへのドラッグでタブを切り替えて、ドロップ

4.4. 属性エリア

各制御アイコンは、固有の情報を持ちます。たとえば、車の移動を表すアイコン は、移動速度や移動時間を情報として持っています。この情報の設定は属性エリアで行うことができます。プログラムエリアに置かれたアイコンを選択すると、選択されたアイコンの属性が属性エリアに表示されますので、意図する設定値を設定することができます。

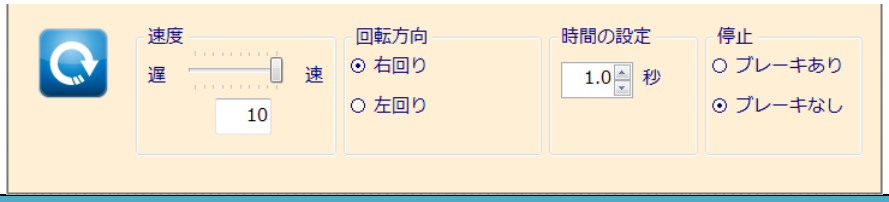
以下に各アイコンとそれぞれが持つ属性を示します。

■ 移動系アイコン１



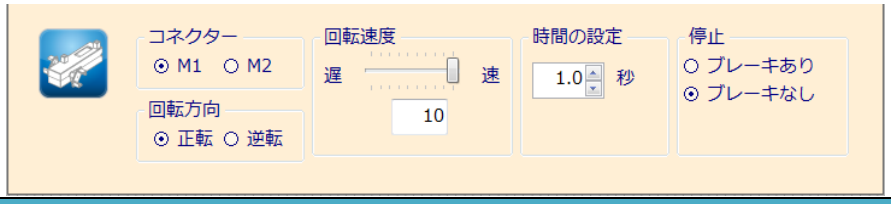
属性	内容
速度	移動速度を 0～10 段階で設定します。
時間の設定	移動時間を 0.1～25.5 秒で設定します。
停止	移動処理後の停止方法を設定します ブレーキありは、その場で停止するため、移動後に停止する位置が重要な処理に向いています。ブレーキなしは、移動処理後、惰性で進み止まります。移動アイコンが次の処理に来る場合、ブレーキなしを選択することで、スムーズに次の移動に移ることができます。

■ 移動系アイコン２



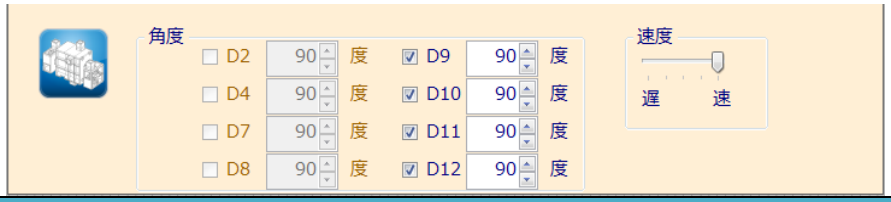
属性	内容
速度	移動速度を 0～10 段階で設定します。
回転方向	右回りか左回りを設定します。
時間の設定	移動時間を 0.1～25.5 秒で設定します。
停止	移動処理後の停止方法を設定します ブレーキありは、その場で停止するため、移動後に停止する位置が重要な処理に向いています。ブレーキなしは、移動処理後、惰性で進み止まります。移動アイコンが次の処理に来る場合、ブレーキなしを選択することで、スムーズに次の移動に移ることができます。

■ DC モーターアイコン



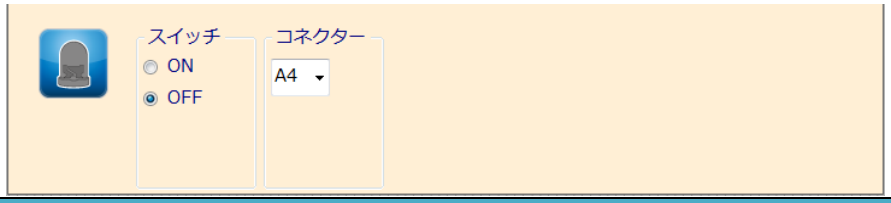
属性	内容
コネクター	対象の DC モーターが基板に接続されているコネクターを設定します。選択可能なコネクターは、**4.5** メインメニューの入出力設定で登録したコネクターになります。
回転方向	DC モーターの正転・逆転を設定します。
回転速度	DC モーターの回転速度を 0～10 段階で設定します。
時間の設定	移動時間を 0.1～25.5 秒で設定します。
停止	DC モーター回転処理後の停止方法を設定します ブレーキありは、その場で停止するため、移動後に停止する位置が重要な処理に向いています。ブレーキなしは、移動処理後、惰性で進み止まります。移動アイコンが次の処理に来る場合、ブレーキなしを選択することで、スムーズに次の移動に移ることができます。

■ サーボモーターアイコン



属性	内容
角度	角度を設定するコネクターをチェックボックスで選択し、角度 0 度～180 度を 1 度単位で設定します。選択可能なコネクターは、**4.5** メインメニューの入出力設定で登録したコネクターになります。
移動速度	サーボモーターの指定角度までの移動速度を 0～20 で設定します。

■ 単色 LED アイコン



属性	内容
スイッチ	LED ON/OFF を設定します。
コネクター	LED が接続されているコネクターを設定します。選択可能なコネクターは、**4.5** メインメニューの入出力設定で登録したコネクターになります。

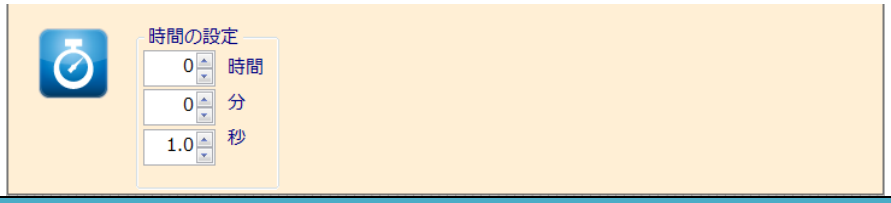
■ ブザーアイコン

属性	内容
出力音	鍵盤で音階を指定し、音の長さを 0.1～1.0 秒で設定します。
コネクター	ブザーが基板に接続されているコネクターを設定します。選択可能なコネクターは、**4.5** メインメニューの入出力設定で登録したコネクターになります。

■ メロディアイコン

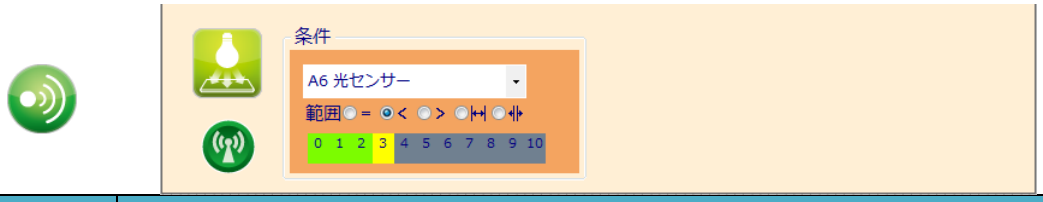
属性	内容
メロディ作成	音符記号と音階(鍵盤)、休符の場合は記号のみを選択し、→ボタンをクリックすることで、音を登録します。また、テンポを 90,120,150 から選択し設定します。 登録可能な音は 8 音までです。
コネクター	ブザーが基板に接続されているコネクターを設定します。選択可能なコネクターは、**4.5** メインメニューの入出力設定で登録したコネクターになります。

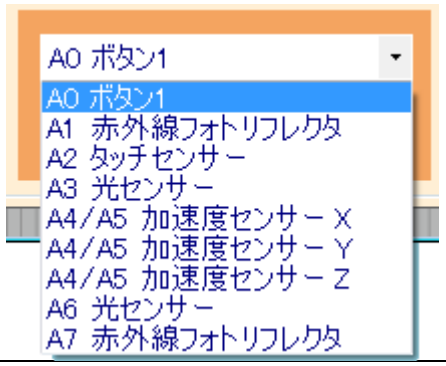
■ ウェイトアイコン



属性	内容
時間の設定	指定された時間が経過するまでウェイト状態になります。時間を 0.1～25.5 秒で設定します。

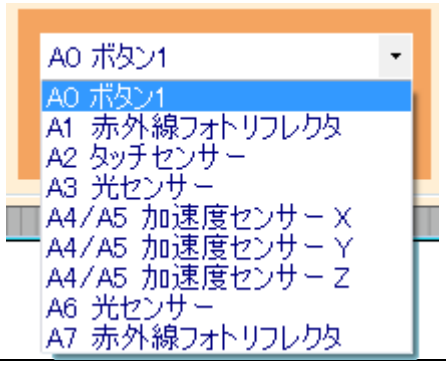
■ センサー条件アイコン



属性	内容
条件	基板に接続されているセンサーをコンボボックスで選択し、条件(等号(=), 不等号(< >), 範囲(|→←|, ←||→)を選択し、条件のしきい値を設定します。範囲は、開区間になります(しきい値は含まれません)。 コンボボックスでは、下図のように基板と接続されているセンサーが表示されます。表示されるセンサーは、**4.5** メインメニューの入出力設定で登録したセンサーになります。 

■ 2センサー条件アイコン

属性	内容
条件	基板に接続されているセンサーをコンボボックスで選択し、条件(等号(=), 不等号(< >), 範囲(|→←|, ←||→)を選択し、条件のしきい値を設定します。範囲は、開区間になります(しきい値は含まれません)。 2 条件間にあるコンボボックスで「かつ」、「または」の論理判定を設定します。 コンボボックスでは、下図のように基板と接続されているセンサーが表示

	されます。表示されるセンサーは、**4.5** メインメニューの入出力設定で登録したセンサーになります。 

4.5. メインメニュー

■ ファイルメニュー

ファイルメニューでは、ファイルの読み込みや保存を行います。

- **開く**

  保存したプログラムを読み込みます。

- **上書き保存**

  ファイルを開いた状態であれば、そのファイルを上書き保存します。

- **名前を付けて保存**

  作成したプログラムに名前を付けて保存します。

- **終了**

  Studuino アイコンプログラミング環境を終了します。

■ 編集メニュー

編集メニューでは、プログラムの作成や編集に関係する機能を提供します。**(項目「オプションパーツを使う」と「赤外線信号登録」は Ver1.x のプログラミング環境では表示されません)**

- **元に戻す**

  一つ前の状態に戻ります。

  ソフトウェアは、アイコンのドロップ・挿入・削除、入出力設定変更時点での状態を、最新のものから最大 10 個まで履歴に保持します。

- **メニュー初期化**

  プログラムエリアに置いたアイコンを初期化します。

- **Arduino 言語を表示**

  プログラムエリアで作成したプログラムを Arduino 言語に変換します。本メニューで出力された Arduino 言語のソースコードは、Arduino IDE でコンパイルし、Studuino へ転送できます。

- **オプションパーツを使う(Ver1.x のプログラミング環境では表示されません)**

  ジャイロセンサー、カラーセンサー、超音波センサー、温度センサー、ブルートゥースモジュールを使用するには、この項目をクリックしてチェックを入れます。各パーツの使用方法については、ウェブサイトのソフトウェアマニュアル「オプションパーツの使用方法」を参照してください。

- **赤外線信号登録(Ver1.x のプログラミング環境では表示されません)**

  赤外線信号登録フォームを起動し、赤外線信号登録モードに移行します。オプションパーツが有効かつ、入出力で赤外線リモコン受信が選択されている必要があります。

- **モーター校正**

  サーボモーターの角度、DC モーターの回転速度のずれを調整し、設定ファイルに書き込みます。このメニューを実行すると、後述のテストモードへと移行し、「モーター校正ダイアログ」が表示されます。

  ➢ **サーボモーター校正**

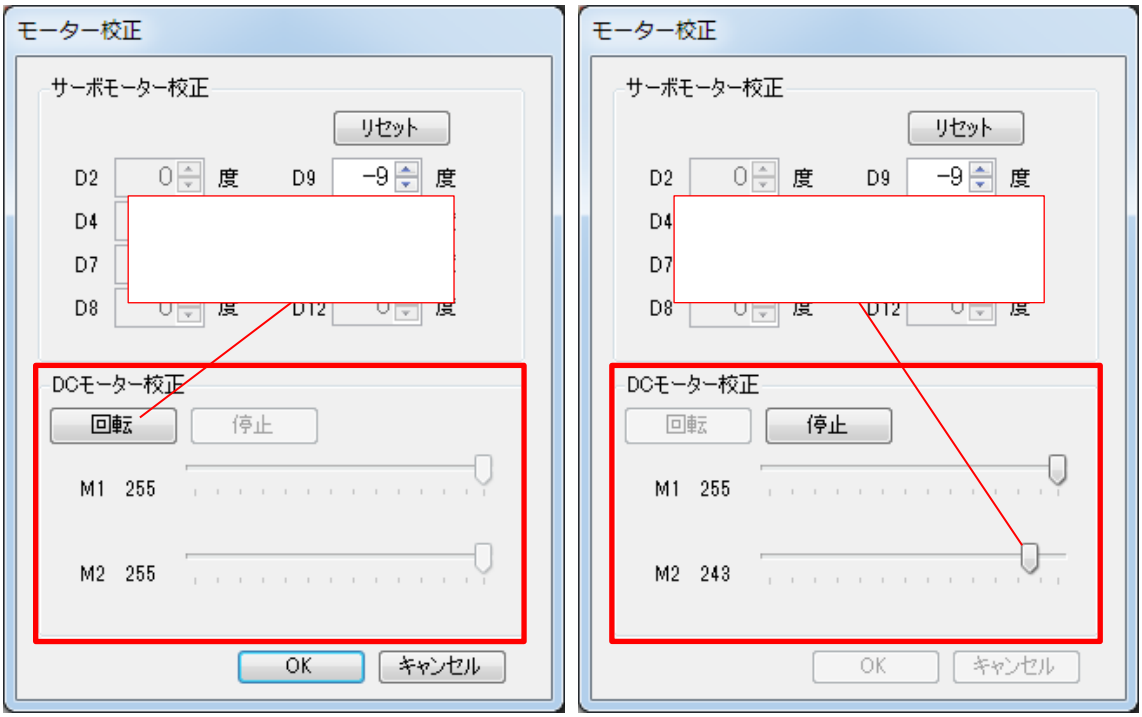
  ダイアログが立ち上がると、接続されているすべてのサーボモーターが 90 度に設定さ

れます。-15~15 までのオフセット値を入力することで、対応するサーボモーターが「90 度+設定したオフセット値」まで動きます。

サーボモーターを見ながら、90 度ピッタリになるオフセット値を探します。

全てのサーボモーターの設定が完了したら、OK ボタンを押します。設定が保存され、次回以降は同じ設定値が使用されます。

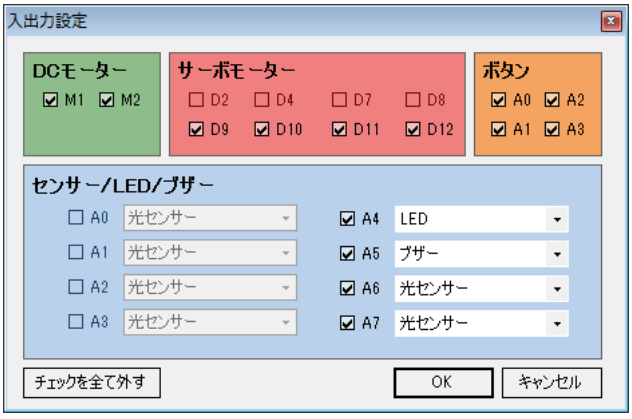
➢ **DC モーター校正**



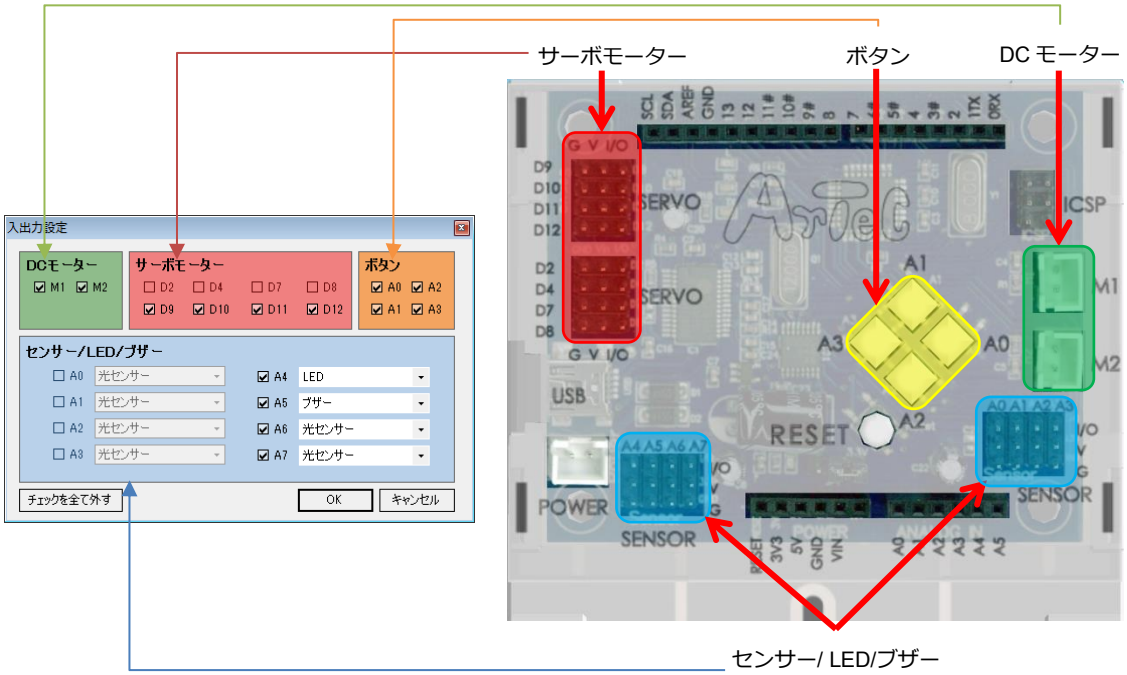
ダイアログ立ち上げ後、回転ボタンを押すと 2 つの DC モーターが最大速度で回転を始めます。トラックバーが有効になるので、つまみを調節して 2 つの DC モーターの回転速度が同じになるようにします。停止ボタンを押すと回転が止まり、OK/キャンセルボタンが再び有効になるので、OK ボタンを押します。設定が保存され、次回以降は同じ設定値が使用されます。

- **入出力設定**

Studuino に接続しているパーツ情報を Studuino アイコンプログラミング環境に登録します。本メニューまたは、アイコンパレットの Operation にある をクリックすることで「入出力設定ダイアログボックス」を表示できます。



このダイアログボックスのチェックボックスは、下記のように Studuino のコネクターに対応していますので、パーツが接続されている Studuino のコネクターに対応するチェックボックスにチェックを入れてください。



また、**1.3 Studuino について**にもあるように、DC モーター用コネクターM1 とサーボモーター用コネクターの D2, D4、DC モーター用コネクターM2 とサーボモーター用コネクターの D7, D8、ボタン(=プッシュスイッチ)A0～A3 とセンサー/LED/ブザー用コネクターA0～A3 は同時に使用できません。この組み合わせは「入出力設定ダイアログボックス」でも同時に設定できなくなっていますので、一方を使用する場合は、片方を無効にしてください。たとえば、サーボモーターの D2, D4 を使用する場合は、DC モーターの M1 のチェックボックスを外すと D2, D4 が有効になり、チェックボックスにチェックを入れることができます。

<注意事項>

コネクターを 2 つ使用する加速度センサーに関しては、対応したコネクター全てにチェックが入っている必要があります。

接続パーツ	コネクターの組み合わせ
加速度センサー	A4,A5

複数コネクターを使用するデバイスのコネクター組み合わせ

上表の「コネクターの組み合せ」に対応するコネクター名のチェックボックスにチェ

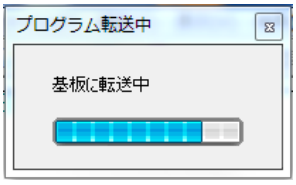
ックを入れた状態で、アイテム選択を行うと、対応する接続パーツが表示され、パーツを選択すると自動的に他のコネクターのアイテムも選択されます。

■ 実行メニュー

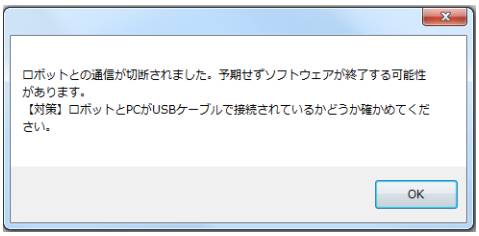
実行メニューでは、Studuino と連携してプログラムを作成する機能を提供します。

● プログラム作成・転送

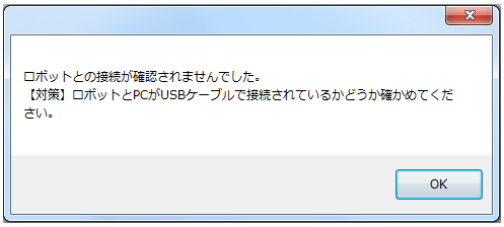
作成したプログラムを Studuino で動作するプログラムにビルドし、ビルドしたプログラムを Studuino に転送します。アイコンパレットとプログラムエリアの間にある をクリックすることで本メニューと同様の処理が実行されます。プログラム作成・転送を開始する前に必ず Studuino と PC が USB 接続されていることを確認して下さい。プログラム作成・転送中は、右のような状況を表すステータスバーが表示されます。プログラム転送後、このステータスバーは非表示になります。プログラム作成・



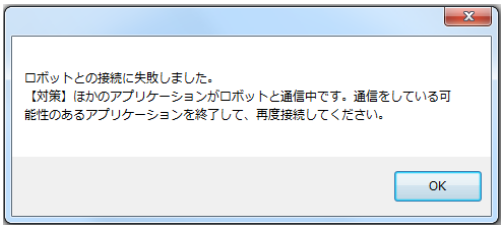
転送時に、Studuino と PC の通信に異常が発生した場合、メッセージを表示します。下記にプログラム作成・転送時に表示されるメッセージを記します。



本メッセージは、プログラム作成・転送中に Studuino と PC の USB 接続が切り離された場合に表示されます。本メッセージが表示された場合、予期せず Studuino アイコンプログラミング環境が終了する可能性がありますので、作成したプログラムを保存し、Studuino アイコンプログラミング環境を再起動して下さい。



本メッセージは、Studuino と PC が USB 接続されていない場合に表示されます。本メッセージが表示された場合、Studuino と PC が USB 接続していることを確認し、再度プログラム作成・転送を実行して下さい。



本メッセージは、Studuino と PC を接続している通信が他のアプリケーションによって使用されている場合に表示されます。Studuino と接続している可能性のあるアプリケーションを終了し、再度プログラム作成・転送を実行して下さい。

- **プログラム実行**

  Studuino に転送したプログラムを実行します。

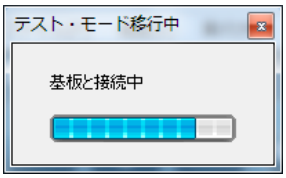
- **テストモード**

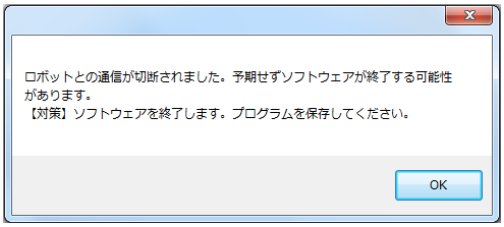
  本メニューまたは、アイコンパレットとプログラムエリアの間にある をクリックすることでテストモードに入ることができます。テストモードでは、Studuino と通信し、Studuino に接続されているパーツをリアルタイムに制御します。テストモードを利用することで、ロボットの動作を確認しながら、アイコンの設定値を決めることができます。テストモード時の各アイコンの動作を下記に記します。

アイコンの種類	確認可能な動作
移動系	プログラムエリアに置いたアイコンをクリックすることで、設定した動きを確

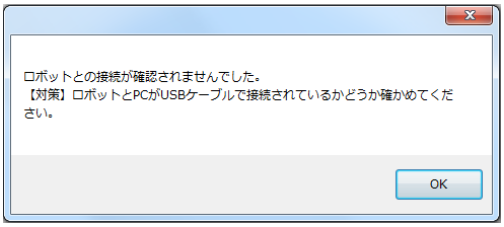
	認することができます。
DC モーター	プログラムエリアに置いたアイコンをクリックすることで、設定した動きを確認することができます。
サーボモーター	プログラムエリアに置いたアイコンをクリックすることで、設定した角度を確認することができます。また、属性エリアで設定する角度もリアルタイムで確認することができます。
LED	属性エリアの ON/OFF 設定に連動して LED が ON/OFF されます。
ブザー	属性エリアで選択された鍵盤の音がブザーで出力されます。
メロディ	プログラムエリアに置いたアイコンをクリックすることで、アイコンに登録されたメロディをブザーで確認することができます。また、属性エリアで選択された鍵盤の音がロボットのブザーで出力されます。

テストモードを開始する前に必ず Studuino と PC が USB 接続されていることを確認して下さい。テストモード移行中は、右のようなテストモード移行状況を表すステータスバーが表示されます。テストモードに移行後、このステータスバーは非表示になります。テストモード移行時に、Studuino と PC の通信に異常が発生した場合、メッセージを表示します。下記にテストモード移行時に表示されるメッセージを記します。

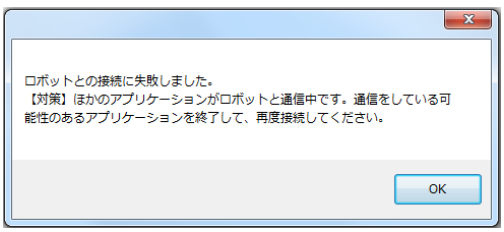




本メッセージは、テストモード移行中または、テストモード時に Studuino と PC の USB 接続が切り離された場合に表示されます。本メッセージが表示された場合、強制的に Studuino アイコンプログラミング環境を終了しますので、表示されるファイル保存ダイアログで作成したプログラムを保存して下さい。



本メッセージは、Studuino と PC が USB 接続されていない場合に表示されます。本メッセージが表示された場合、Studuino と PC が USB 接続していることを確認し、再度テストモードを実行して下さい。

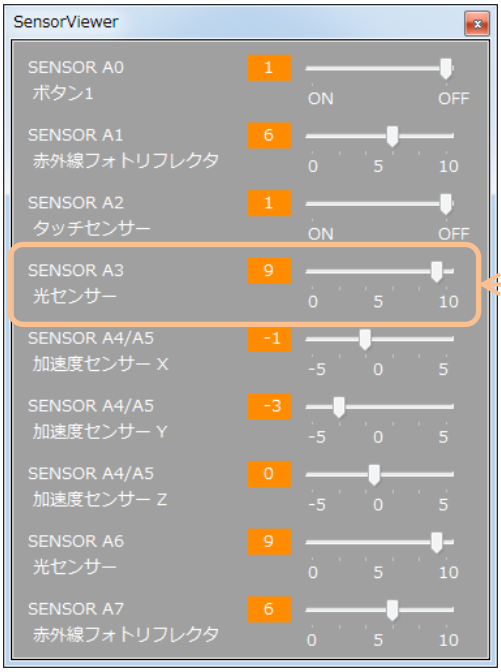


本メッセージは、Studuino と PC を接続して

いる通信が他のアプリケーションによって使用されている場合に表示されます。Studuino と接続している可能性のあるアプリケーションを終了し、再度テストモードを実行して下さい。

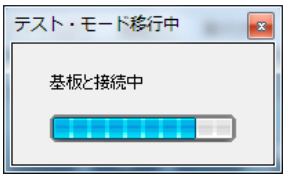
- **センサー値確認モード**

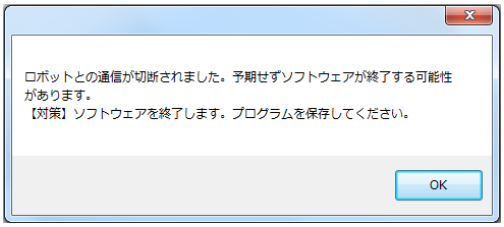
本メニューまたは、センサーアイコンの属性エリアにある をクリックすることでセンサー値確認モードに入ることができます。Studuino と通信し、センサービューアを通して、Studuino に接続されているセンサーパーツからの入力値を表示します。



接続されているセンサーの
値を表示します。

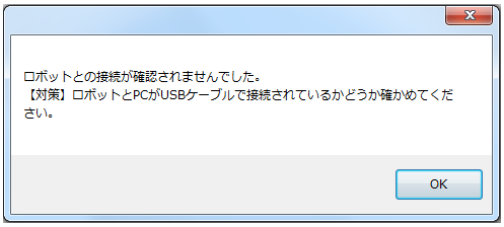
センサー値確認モードを開始する前に必ず Studuino と PC が USB 接続されていることを確認して下さい。センサー値確認モード移行中は、右のようなテストモード移行状況を表すステータスバーが表示されます。センサー値確認モードに移行後、このステータスバーは非表示になります。センサー値確認モード移行時に、Studuino と PC の通信に異常が発生した場合、メッセージを表示します。下記にセンサー値確認モード移行時に表示されるメッセージを記します。



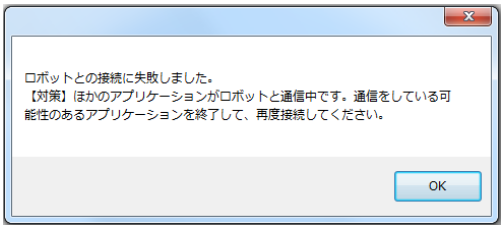


本メッセージは、センサー値確認モード移行中または、センサー値確認モード時に Studuino と PC の USB 接続が切り離された

場合に表示されます。本メッセージが表示された場合、強制的に Studuino アイコンプログラミング環境を終了しますので、表示されるファイル保存ダイアログで作成したプログラムを保存して下さい。



本メッセージは、Studuino と PC が USB 接続されていない場合に表示されます。本メッセージが表示された場合、Studuino と PC が USB 接続していることを確認し、再度センサー値確認モードを実行して下さい。



本メッセージは、Studuino と PC を接続している通信が他のアプリケーションによって使用されている場合に表示されます。Studuino と接続している可能性のあるアプリケーションを終了し、再度センサー値確認モードを実行して下さい。

■ ヘルプメニュー

ヘルプメニューでは、Studuino プログラミング環境についての情報を提供します。

- **Studuino アイコンプログラミング環境について**

  アイコンプログラミング環境についての情報を表示します。

4.6. コンテキストメニュー

プログラミングエリアの変更したいボックス上で右クリックをすると表示され、コンテキストメニューが表示されます。

- **消去**

  アイコンを消去します。

- **削除**

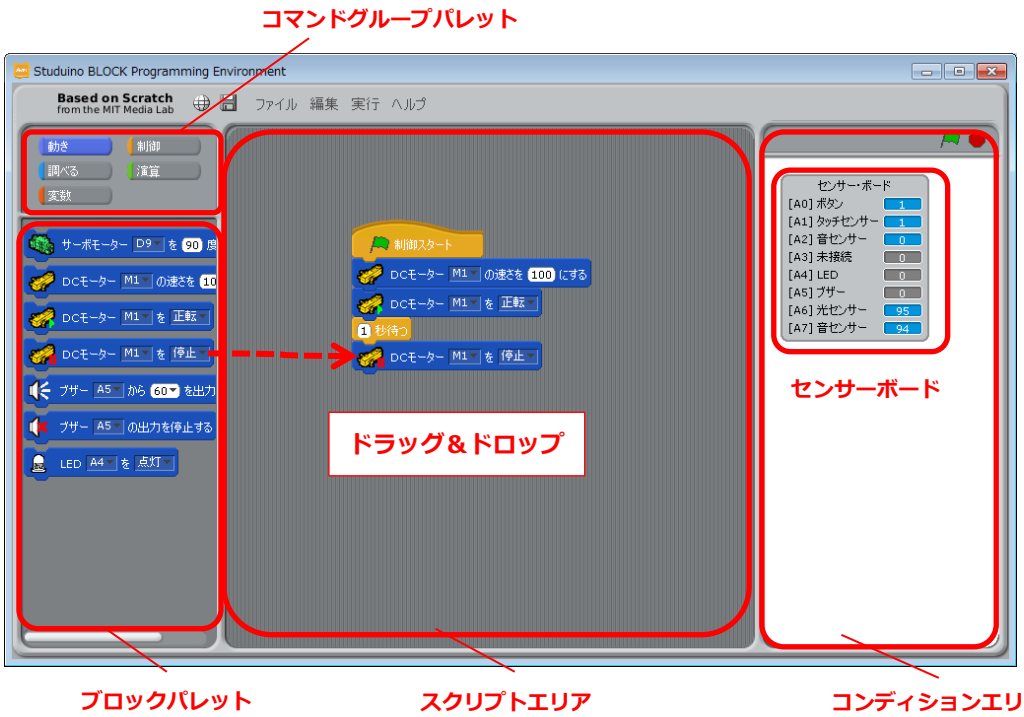
  処理を消去し、左詰めします。

- **挿入**

  空の処理を挿入します

5. Studuino (スタディーノ) ブロックプログラミング環境

5.1. 概要・特徴



Studuino ブロックプログラミング環境は、マサチューセッツ工科大学が開発した教育用のプログラミング環境 Scratch(スクラッチ)をベースとして、ロボット制御用に開発したビジュアルプログラミング環境です。ブロックパレットに置かれているブロックをスクリプトエリアにドラッグ＆ドロップで移動し、他のブロックと繋げていくことでロボットを制御するプログラムを作成することができます。

使用できるブロックは、プログラミングの基本要素である分岐や繰り返し、変数に対応しているため、プログラミング言語(例えば C 言語)と同等のプログラムを作成することができます。また、テストモードを使用する事で、リアルタイムにロボットと通信することができます。テストモードでは、センサーの値を確認したり(センサーボード)、ロボットの状態を確認しながらプログラムを作成することができます(**5.5 メインメニュー**を参照)。

プログラムが完成したら、プログラム作成・転送メニューでプログラムを Studuino へ転送し実行することができます。また、作成した制御プログラムを Arduino(アルドゥイーノ)言語に変換し、Arduino 言語でより高度な制御プログラムを作成することもできます(**5.5 メインメニュー**の Arduino 言語に変換を参照)。

以下に Studuino ブロックプログラミング環境の詳細を記します。

5.2. コマンドグループパレットとブロックパレット

コマンドグループパレットとブロックパレットでは、プログラム作成に使用するブロックを管理します。コマンドグループパレットのボタンを選択することで、ブロックパレットの表示が切り替わります。下記にコマンドグループパレットのボタンと対応するブロックパレットで表示されるブロックを記します。

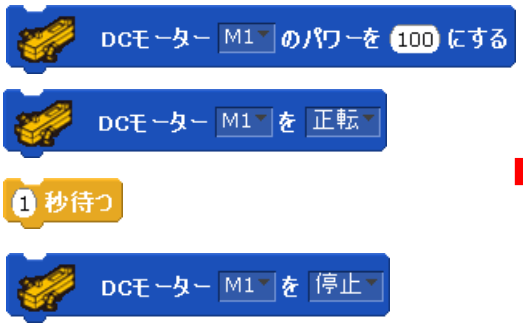
ボタン	ブロック
動き	DC モーター、サーボモーター、ブザー、LED を制御するブロック
制御	プログラミングの基本要素である、分岐や繰り返し、関数やウェイト処理ブロック
調べる	センサーの値を取得するブロック
演算	四則演算、算術演算、論理演算ブロック
変数	変数とその変数に対する処理ブロック、リストとそのリストに対する処理ブロック

5.2.1. ブロックについて

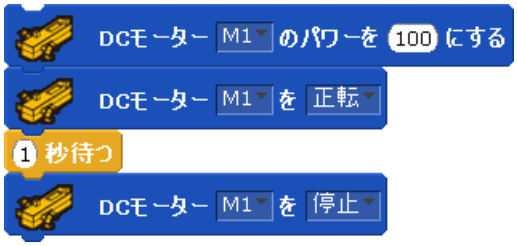
Studuino ブロックプログラミング環境では、ブロックの種類は、

のような上下に凹凸があるブロック(処理ブロック)と

や
のような上下に凹凸がないブロックに分けられます(設定ブロック)。処理ブロックは、主に処理を表し、処理ブロックどうしを繋げることでロボットを制御するプログラムを作成します



処理ブロックどうしを上下につなぐ

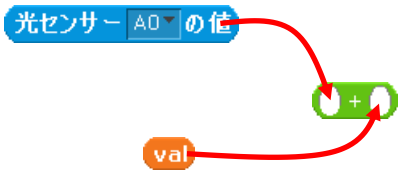


DC モーターの速度を 100 に設定し、
1 秒間回転した後に停止する処理

のような、ブロックの端が丸いブロックは、値を返すブロックで、主に他のブロックの設定として使用されます。
のような、六角形のブロックは、条件を判定するブロックで、主に

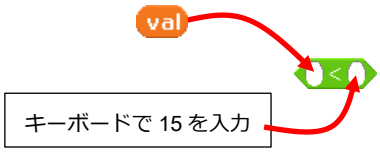
のような条件処理ブロックの条件部分に使用されま

す。



光センサーの値ブロックと val ブロックを
加算ブロックに設定する

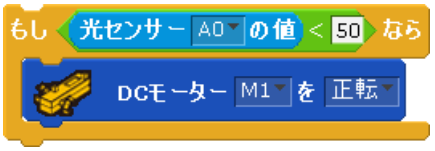
光センサーと val を加算する
ブロックを表す



val ブロックと 15(キーボード入力)を
比較演算＜ブロックに設定する

val は 15 よりも小さいかを
判定するブロックを表す

光センサーの値が 50 よりも小さいかを
判定するブロックと DC モーターを正転
するブロックを分岐ブロックに設定する



もし、光センサーの値が 50 よりも小さいな
らば、DC モーターを正転する処理を表す

またブロックへの設定は、下の①のような端が丸い入力部分と②のような六角形のくぼみがあります。

①の端が丸い入力部分には、

のような、ブロックの端が丸いブロックまたは、数字が設定できます。②のような六角形のくぼみには、

のような、六角形のブロックが設定できます。以下に各ブロックパレットで表示されるブロックについて記します。

■　「動き」パレットで提供されるブロック

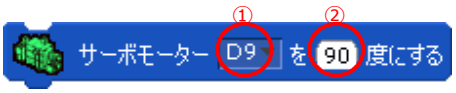
「動き」パレットでは、ロボットのパーツを制御するブロックを提供します。入出力設定されていないパーツを示すブロックは、下記のように灰色で表示され、スクリプトエリアへのドラッグ＆ドロップができなくなります。

Studuino と接続されている場合　　Studuino と接続されていない場合

以下に各ブロックについて記します。

- **サーボモーター制御ブロック**

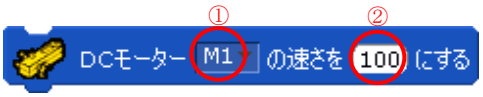
  基板に接続されているサーボモーターを制御するブロックです。



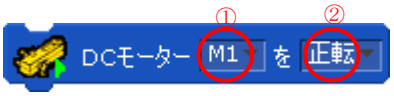
  ①で指定したコネクター(D2～D12)に接続されているサーボモーターを、②で指定した角度(0 度～180 度)に設定します。0 度よりも小さい値が設定された場合は 0 度に、180 度よりも大きい値が設定された場合は 180 度に設定されます。

- **DC モーター制御ブロック**

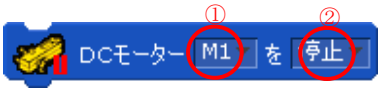
  基板に接続されている DC モーターを制御するブロックです。



  ①で指定したコネクター(M1/M2)に接続されている DC モーターを、②で指定した速さ(0～100)に設定します。速さは大きいほど DC モーターの回転速度が速くなります。0 よりも小さい値が設定された場合は 0 に、100 よりも大きい値が設定された場合は 100 に設定されます。

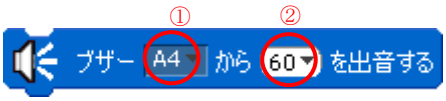


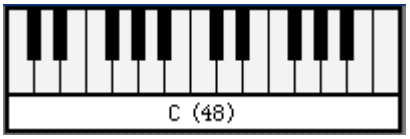
  ①で指定したコネクター(M1/M2)に接続されている DC モーターを、②で指定した方向(正転/逆転)でスタートします。



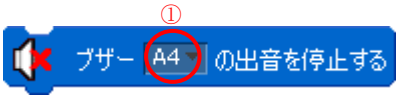
  ①で指定したコネクター(M1/M2)に接続されている DC モーターを、②で指定した停止方法(停止/解放)で停止します。

- **ブザー制御ブロック**



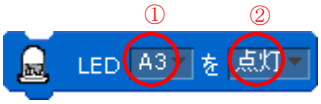


①で指定したコネクター(A0～A5)に接続されているブザーを②で指定した音の高さで音を出力します。②の▼をクリックし、表示される鍵盤でブザーから鳴らしたい音階の番号を選択します。選択可能な音階は、48(C3, 130Hz)～107(C8 , 4186Hz)までになります。鍵盤では、48(C3, 130Hz)～72(C5, 523Hz)まで選択できます。72 以上の音階の番号を設定する場合は、数字を直接入力して下さい。また、音を出力している間は、M1 コネクターに接続された DC モーターの制御は保証していません。DC モーター制御ブロックで M1 コネクターに接続された DC モーターを制御する場合は、下記の音の出力を停止するブロックを使用してから DC モーター制御ブロックを使用して下さい。



①で指定したコネクター(A0～A5)に接続されているブザーの出力を停止します。

- **LED 制御ブロック**



①で指定したコネクター(A0～A5)に接続されている LED を、点灯/消灯します。

■ 「制御」パレットで提供されるブロック

「制御」パレットでは、プログラム制御フローを表すブロックを提供します。以下に各ブロックについて記します。

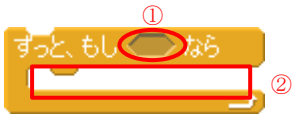
関数を開始します。①に関数名指定します。関数名に英数字以外を設定するとプログラム作成・転送時にエラー「接続されていないブロックが存在します」が出力されます(5.5 メインメニューのプログラム作成・転送を参照)。関数名は英数字のみ使用して下さい。

①で指定した関数を実行します。

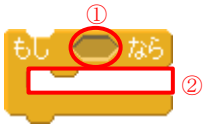
①で指定した秒数ウェイトします。

①に挿入した処理を永遠に繰り返し実行します。

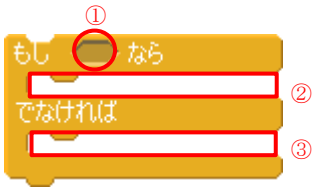
①に挿入した処理を②で設定した回数分繰り返し実行します。



永遠に①の条件が成立するかどうかを評価し、成立した場合、②に挿入した処理を実行します。①の条件は、演算パレットのブロックで指定します。



①の条件が成立した場合、②に挿入した処理を実行します。①の条件は、演算パレットのブロックで指定します。



①の条件が成立した場合、②に挿入した処理を実行します。成立しなかった場合、③に挿入した処理を実行します。①の条件は、演算パレットのブロックで指定します。

①で指定した条件が成立するまでウェイトします。①の条件は、演算パレットのブロックで指定します。

②に挿入した処理を①で指定した条件が成立するまで繰り返し実行します。①の条件は、演算パレットのブロックで指定します。

②に挿入したサーボモーター制御ブロックを①で指定した速さ移動させます。本ブロックに挿入できるブロックはサーボモーター制御ブロックのみです。サーボモーター制御ブロック以外のブロックを挿入した場合、動作の保証はされません。

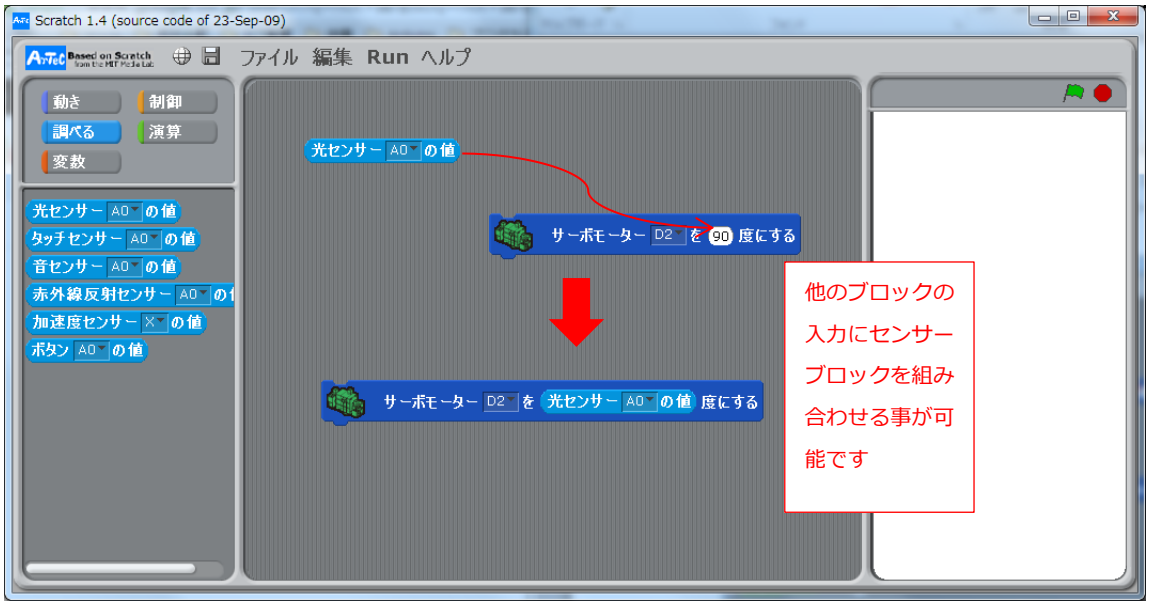
■　「調べる」パレットで提供されるブロック

「調べる」パレットでは、センサーの値を取得するブロックを提供します。入出力設定されていないパーツは、下記のように灰色で表示され、スクリプトエリアへのドラッグ＆ドロップができなくなります。

Studuino と接続されている場合

Studuino と接続されていない場合

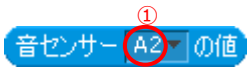
センサーブロックは、他のブロックの設定値に組み合わせる事が可能です。下図の例は、サーボモーターブロックと光センサーブロックを組み合わせる事で、サーボモーターの角度を明るさで変える処理を表しています。



以下に各ブロックについて記します。

①で指定したコネクター(A0～A7)に接続されている光センサーの値を返します。センサー値は、0～100 で表されます。

①で指定したコネクター(A0～A5)に接続されているタッチセンサーの値を返します。センサー値は、ボタンを押している状態は 0、離している状態は 1 で表されます。



①で指定したコネクター(A0～A7)に接続されている音センサーの値を返します。センサー値は、0～50 で表されます。

①で指定したコネクター(A0～A7)に接続されている赤外線反射センサーの値を返します。センサー値は、0～100 で表され

ます。

加速度センサーの座標軸(X/Y/Z)方向の値を返します。センサー値は、0～100 で表されます。

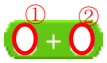
①で指定したコネクター(A0～A3)に接続されているプッシュスイッチの値を返します。スイッチを押している状態は 0、離している状態は 1 で表されます。

タイマーの値を 0 にリセットします

タイマーの値を返します。単位は秒です。

■　「演算」パレットで提供されるブロック

「演算」パレットでは、入力値に対する演算処理を行うブロックを提供します。以下に各ブロックについて記します。



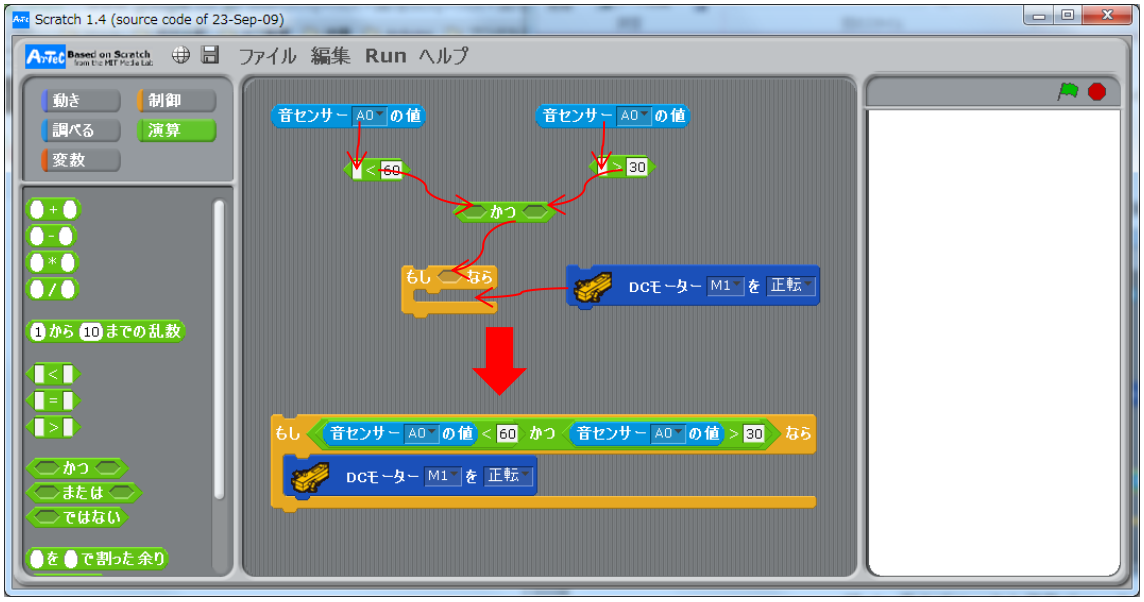
①に設定した値と②に設定した値の加算結果を返します。四則演算ブロックとして、このブロックの他に減算(-)、乗算(*)、除算(/)ブロックがあります。ブロック上で右クリックすることでコンテキストメニューから四則演算を選択することができます。

①に設定した値と②に設定した値の間の乱数を返します。

①に設定した値が②に設定した値より小さいかどうかの判定結果を返します。比較演算ブロックとして、このブロックの他に等号(=)、大なり(>)ブロックがあります。ブロック上で右クリックすることでコンテキストメニューから比較演算子を選択することができます。

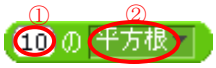
下図では、光センサーと条件ブロック、サーボモーターブロックを組み合わせる事で、明るさがしきい値(50)よりも暗い場合に、サーボモーターを 90 度に設定する処理を表しています。

①に設定した条件と②に設定した条件の論理積の判定結果を返します。論理演算ブロックとして、このブロックの他に論理和(または)、否定(ではない)ブロックがあります。ブロック上で右クリックすることでコンテキストメニューから論理演算子を選択することができます。下図では、音センサー、<判定ブロック、条件ブロック、DC モーターブロックを組み合わせることで、音がしきい値の範囲内(30～60)ならば、DC モーターを正転で回転させる処理を表しています。



①に設定した値と②に設定した値の剰余を返します。演算ブロック と同様、ブロック上で右クリックすることでコンテキストメニューから四則演算を選択することができます。

①に設定した値の小数点以下を丸めた結果を返します。



①に設定した値に②に指定した算術演算の結果を返します。指定できる演算は、絶対値、平方根、三角関数、対数、指数です。

■　**「変数」パレットで提供されるブロック**

「変数」パレットでは、変数とリストを作成できます。変数を作成する場合は、「新しい変数を作る」ボタンをクリックし、変数名を記入することで変数ブロックを作成することができます。また、リストを作成する場合は、「リストを作る」ボタンをクリックし、リスト名を記入することでリストを作成することができます。変数名、リスト名ともに英数字以外を設定するとプログラム作成・転送時にエラー「接続されていないブロックが存在します」が出力されます(5.5 メインメニューのプログラム作成・転送を参照)。変数名、リスト名は英数字のみ使用して下さい。変数、またはリストに追加できる変数の使用可能な値の範囲は-3.4028235E+38 から 3.4028235E+38 までで、32 ビット(4 バイト)のサイズです。

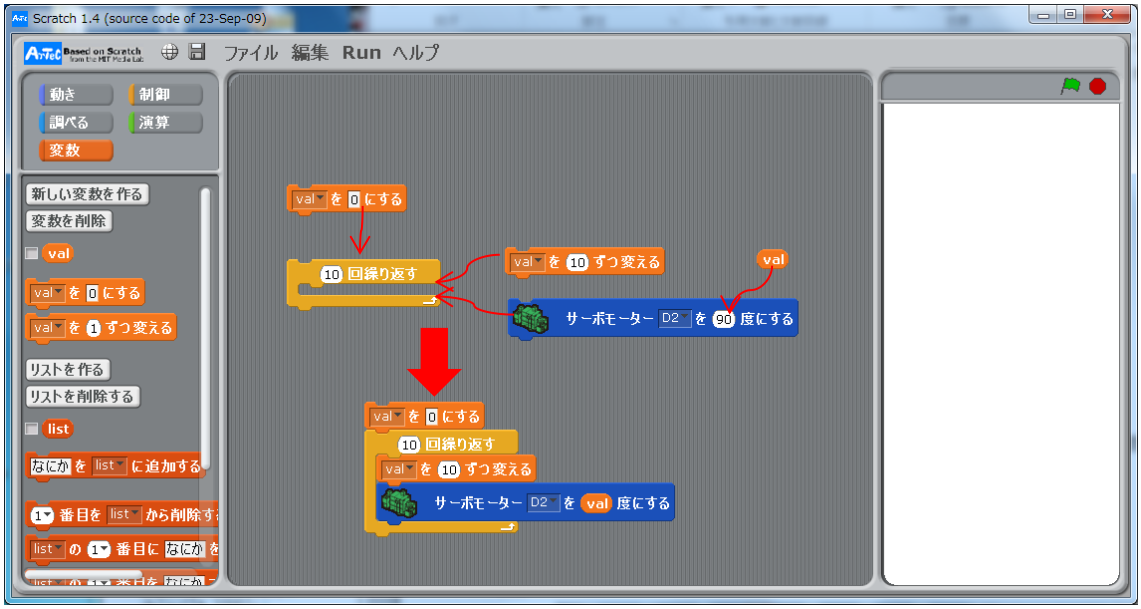
以下に変数を作成した際に表示されるブロックを記します(変数名を val としています)。

変数の値を返します

①で指定した変数を②で指定した値に設定します

①で指定した変数に②で指定した値を加算します。

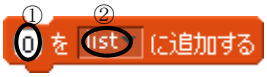
作成可能な変数の数は、最大で 70 個までです。



上図は、「繰り返しブロック」と「サーボモーターブロック」とを組み合わせたプログラムです。変数 val の値を 10 増やし、その値をサーボモーターに設定する処理を 10 回繰り返すことで、サーボモーターを 10～100 度の間で 10 度ずつ設定します。

リストブロックは、リストの好きな位置に値を追加・削除できる構造をもったブロックです。リストに追加可能な値の数は、最大で 40 個までです。以下にリスト名を list として作成したリストブロックを示します(変数名を list としています)。

リストの先頭の値を返します



①で指定した値を②で指定したリストに設定します

②で指定したリストの①番目の要素を削除します

①で指定したリストの②番目に③で指定した値を挿入します

①で指定したリストの②番目に③で指定した値を設定します

①で指定したリストの②番目の値を返します

①で指定したリストの長さを返します

①で指定したリストに②で指定した値が含まれるかどうかの判定結果を返します

上図では、最初の 3 つのブロックで list に 10、20、30 の値を追加します。この処理により、list は、コンディションエリアにあるような 1 番目に 10、2 番目に 20、3 番目に 30 の値が入ったリストになります。このリストの各値を使用して、続くブロックでサーボモーターの角度を設定しています。D9 に接続されたサーボモーターは 10 度に、D10 に接続されたサーボモーターは 20 度に、D11 に接続されたサーボモーターは 30 度に設定されます。

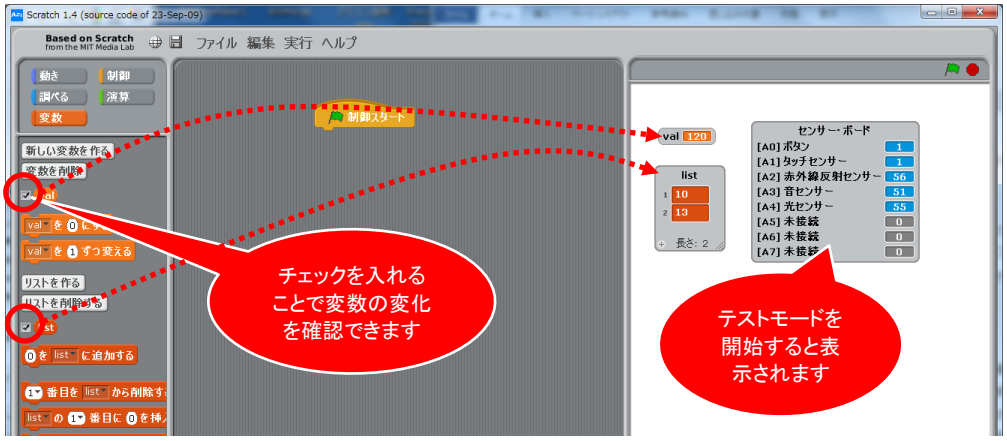
5.3. スクリプトエリア

スクリプトエリアでは、ブロックをドラッグ＆ドロップし、接続することでプログラムを作成します。Studuino ブロックプログラミング環境を起動すると、スクリプトエリアに制御スタートブロック

が表示されます。このブロックは制御プログラムの開始を表します。作成するプログラムは、必ずこのブロックに接続して下さい。

5.4. コンディションエリア

コンディションエリアでは、テストモード時のセンサーの値や、変数やリストの値を表示します。テストモードが開始されると、センサーボードを表示し、Studuino に接続されているセンサーの値をリアルタイムに表示します。また、変数やリストを作成した際に作成される変数ブロックまたはリストブロックの左に表示されるチェックボックスにチェックを入れることにより、変数またはリストの値の変化を確認することができます。



5.5. メインメニュー

■ ファイルメニュー

ファイルメニューでは、プロジェクトの読み込みや保存を行います。

- **新規**

  新規プロジェクトを作成します。

- **開く...**

  保存したプロジェクトを開きます。

- **保存する**

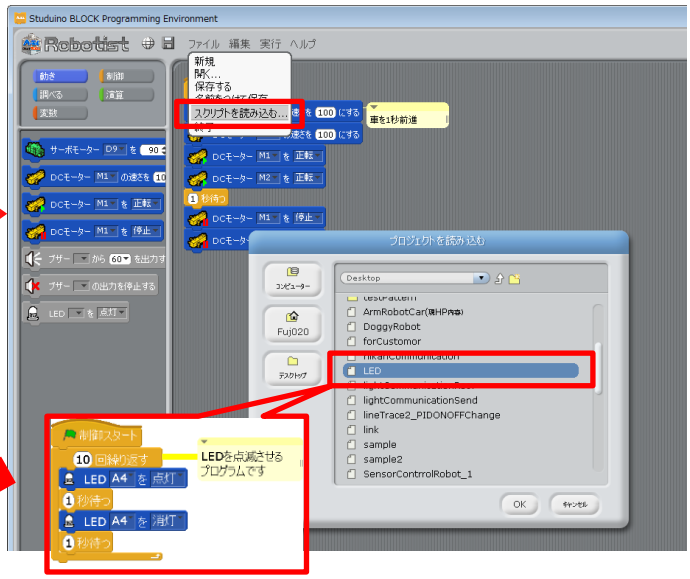
  プロジェクトを上書き保存します。

- **名前を付けて保存...**

  プロジェクトを名前を付けて保存します。

- **スクリプトを読み込む...**

  プロジェクトファイルからスクリプトを読み込みます。



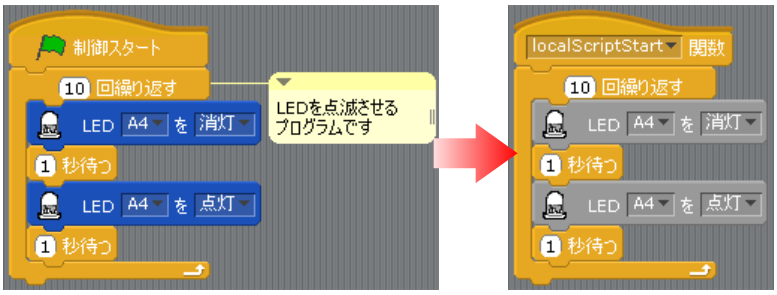
「スクリプトを読み込む」を選択し表示されるダイアログで
読み込みたいプロジェクトファイルを選択

スクリプトが追加されます

読み込んだスクリプトは下記の３点のようになります。

１. プログラミング環境の性質上、「制御スタート」ブロック(緑色の旗が付いたブロック)は 2 つ以上表示できないため、読み込んだプログラムは、localScriptStart 関数に変更されます。

２. 読み込まれたプログラムのコメントは削除されます。



プロジェクトファイルのスクリプト　　読み込まれたスクリプト

3．入出力設定は読み込んだ側の設定になります。読み込んだスクリプトが未設定のパーツブロックを使用している場合は灰色表示になります。入出力設定ダイアログでパーツを設定することで有効にできます。

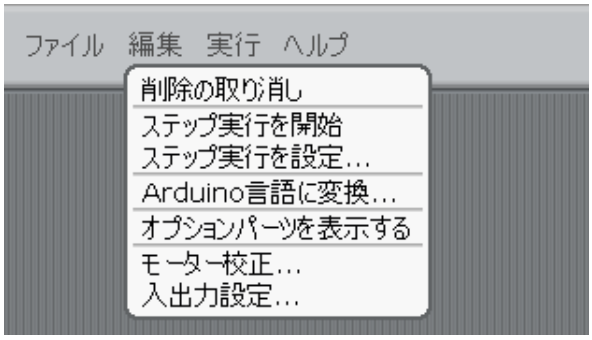
スクリプトを読み込んだ側でパーツを設定する

- **終了**

  Studuino ブロックプログラミング環境を終了します。

■ 編集メニュー

編集メニューでは、プログラムの作成や編集に関係する機能を提供します。**(項目「オプションパーツを表示する」は Ver1.x のプログラミング環境では表示されません)**



- **削除の取り消し**

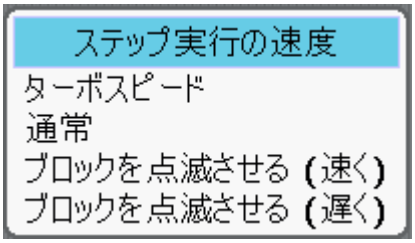
  削除したブロックを元に戻します。

- **ステップ実行を開始/中止**

  ステップ実行を開始します。ステップ実行開始後にブロックを実行すると、実行中のブロックが黄色で表示されます。ステップ実行の速度は、「ステップ実行を設定...」メニュー項目で行います。

- **ステップ実行を設定...**

  ステップ実行時の速度を設定します。ターボスピードが最も速く、ブロックを点滅させる(遅く)が最も遅いステップ処理になります。



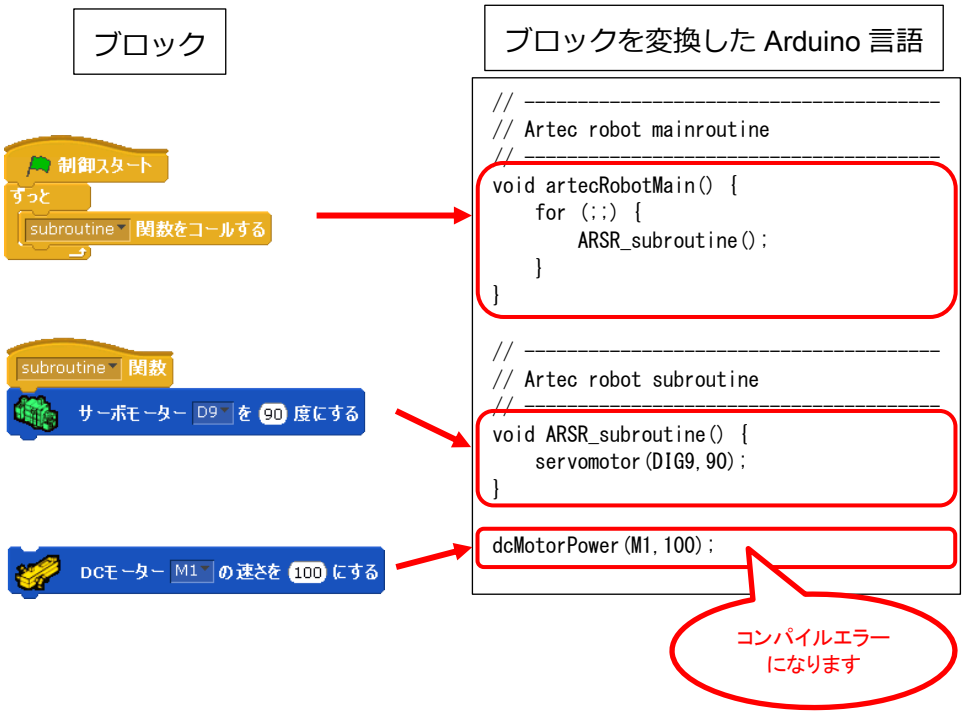
- **Arduino 言語に変換...**

  スクリプトエリアに作成したプログラムを Arduino 言語に変換します。スクリプトエリアに置かれているブロック全てを Arduino 言語に変換します。本メニューで出力された Arduino 言語のソースコードは、Arduino IDE でコンパイルし、Studuino へ転送することが可能です。

  メイン関数ブロック

やサブ関数ブロック

に繋がれていないブロックも Arduino 言語に変換しますので、本メニューで Arduino 言語に変換したファイルを Arduino IDE でコンパイルする際には、関数ブロックに繋がれていないブロックのコードを削除してからコンパイルして下さい。



関数ブロックに繋がっているブロックは、関数内に展開されますが、関数ブロックに繋がれていないコードは、グローバルに展開されるため、コンパイル時にエラーとなります。また、 ブロックのように値を設定するブロックに値が設定されていない場合は、0 を設定して Arduino 言語に変換します。 ブロックのように条件を設定するブロックに条件が設定されていない場合は、false を設定して Arduino 言語に変換します。

- **オプションパーツを表示する(Ver1.x のプログラミング環境では表示されません)**

  超音波センサー、赤外線リモコン受信センサー、温度センサー、ジャイロセンサー、カラーセンサー、Bluetooth を制御するブロックを「調べる」パレットに表示します。

- **オプションパーツを隠す(Ver1.x のプログラミング環境では表示されません)**

  「オプションパーツを表示する」を選択した場合に表示されるメニューです。「オプションパーツを表示する」で「調べる」パレットに表示されたブロックを非表示にします。

各オプションパーツとオプションパーツを制御するブロックについての説明は、下記ウェブサイトにある、「オプションパーツの使用方法」の資料を参照してください。

http://www.artec-kk.co.jp/studuino/Studuino_dl.html

- **モーター校正**

  サーボモーターの角度、DC モーターの回転速度のずれを調整し、設定ファイルに書き込みます。このメニューを実行すると、後述のテストモードへと移行し、「モーター校正ダイアログ」が表示されます。ver1.x のプログラミング環境では、下記のダイアログボックスでモーター校正を行います。

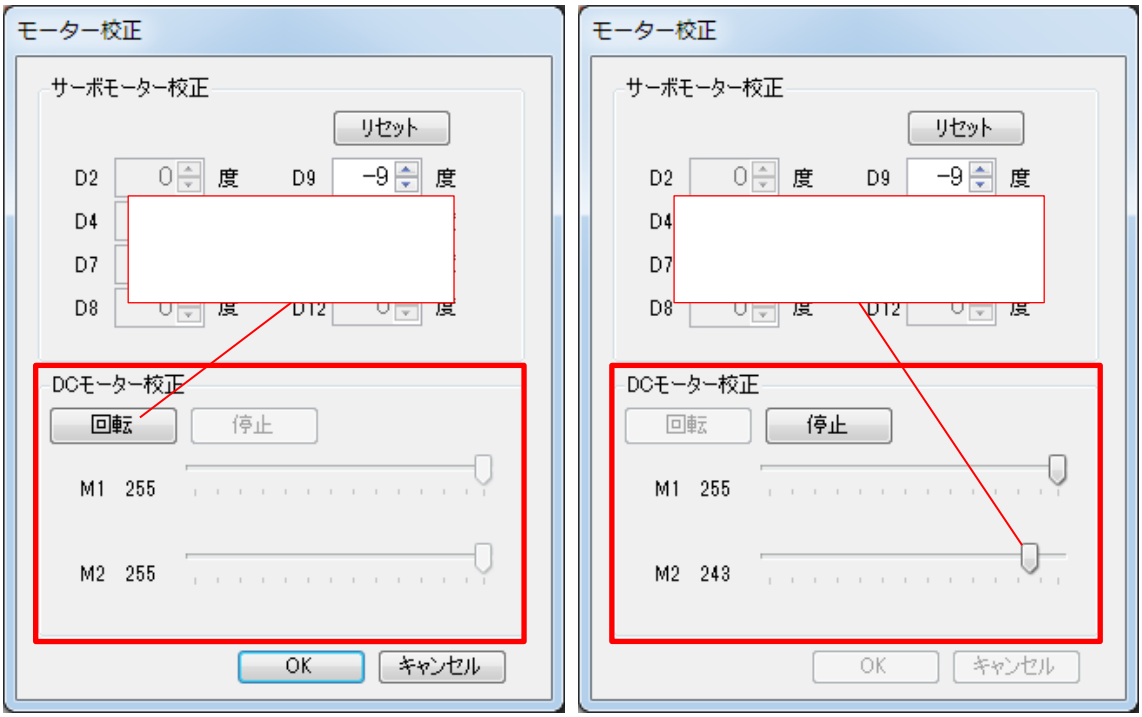
  - **サーボモーター校正**

  ダイアログが立ち上がると、接続されているすべてのサーボモーターが 90 度に設定されます。-15~15 までのオフセット値を入力することで、対応するサーボモーターが「90 度+設定したオフセット値」まで動きます。

  サーボモーターを見ながら、90 度ピッタリになるオフセット値を探します。

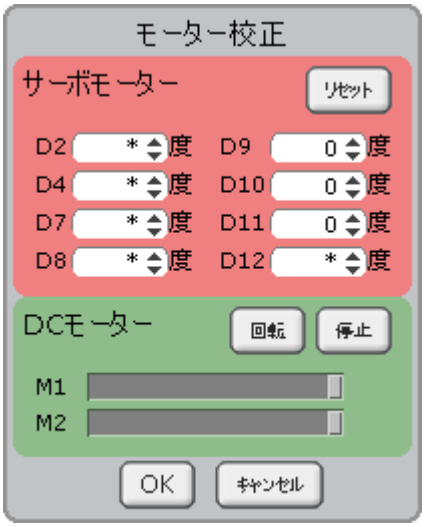
  全てのサーボモーターの設定が完了したら、OK ボタンを押します。設定が保存され、次回以降は同じ設定値が使用されます。

➢ **DC モーター校正**



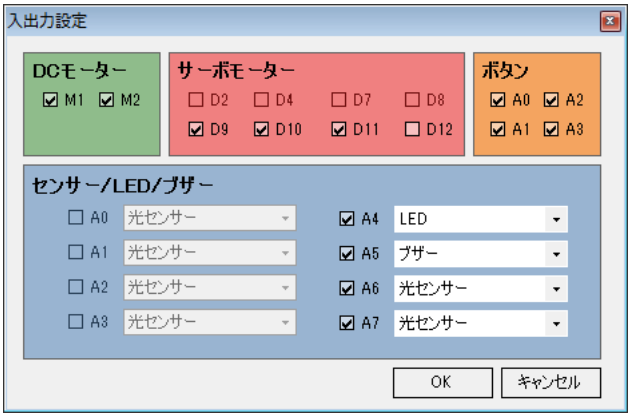
ダイアログ立ち上げ後、回転ボタンを押すと 2 つの DC モーターが最大速度で回転を始めます。トラックバーが有効になるので、つまみを調節して 2 つの DC モーターの回転速度が同じになるようにします。停止ボタンを押すと回転が止まり、OK/キャンセルボタンが再び有効になるので、OK ボタンを押します。設定が保存され、次回以降は同じ設定値が使用されます。

ver2.x のプログラミング環境では、下記のダイアログボックスでモーター校正を行います。ボタンやスライダーは、ver1.x と同様に使用します。

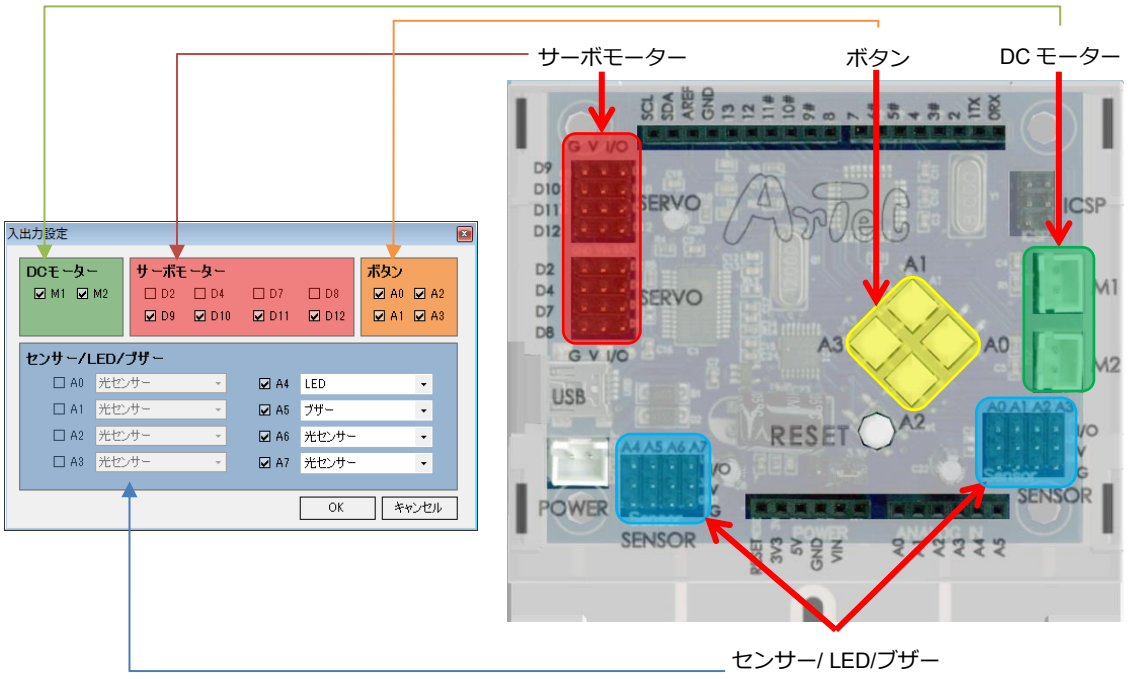


- **入出力設定...**

  Studuino に接続しているパーツ情報を Studuino アイコンプログラミング環境に登録します。このメニューを実行すると「入出力設定ダイアログボックス」が表示されます。



このダイアログボックスのチェックボックスは、下記のように Studuino のコネクターに対応していますので、パーツが接続されている Studuino のコネクターに対応するチェックボックスにチェックを入れてください。



また、**1.3 Studuino について**にもあるように、下記コネクターの組合せは同時に使用できません。

- DC モーター用コネクターM1 とサーボモーター用コネクターの D2, D4
- DC モーター用コネクターM2 とサーボモーター用コネクターの D7, D8
- ボタン(=プッシュスイッチ)A0～A3 とセンサー/LED/ブザー用コネクターA0～A3

この組み合わせは「入出力設定ダイアログボックス」でも同時に設定できなくなっていますので、一方を使用する場合は、片方を無効にしてください。たとえば、サーボモーターの D2, D4 を使用する場合は、DC モーターの M1 のチェックボックスを外すと D2, D4 が有効になり、チェックボックスにチェックを入れることができます。

<注意事項>

コネクターを 2 つ使用する加速度センサーに関しては、対応したコネクター全てにチェックが入っている必要があります。

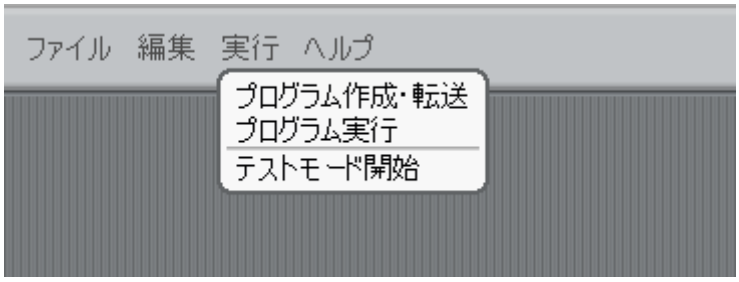
接続パーツ	コネクターの組み合わせ
加速度センサー	A4,A5

複数コネクターを使用するデバイスのコネクター組み合わせ

上表の「コネクターの組み合せ」に対応するコネクター名のチェックボックスにチェックを入れた状態で、アイテム選択を行うと、対応する接続パーツが表示され、パーツを選択すると自動的に他のコネクターのアイテムも選択されます。

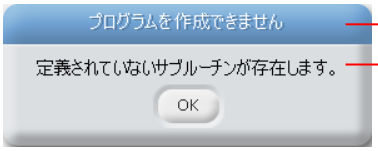
■ 実行メニュー

実行メニューでは、Studuino と連携してプログラムを作成する機能を提供します。



- **プログラム作成・転送**

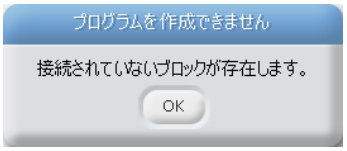
  作成したプログラムを Studuino で動作するプログラムにビルドし、ビルドしたプログラムを Studuino に転送します。ビルドまたは転送時にエラーが発生した場合は、下図のようなメッセージボックスが表示されます。



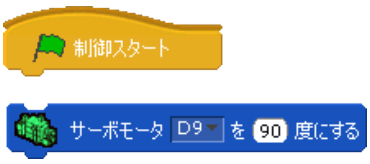
メイン関数ブロック

か関数ブロック

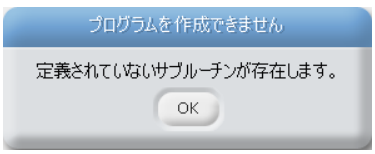
に繋がれていないブロックがスクリプトエリアにある場合、ビルドエラーになります。下記にビルドエラー時に表示されるメッセージボックスとエラーの内容を記します。



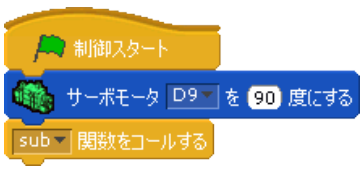
本メッセージは、下図の例のように、メイン関数ブロックか関数ブロックに繋がれていないブロックが存在する場合に表示されます。ビルドを通す場合は、関数ブロックに繋がれていないブロックを削除するか、関数ブロックに繋げて、プログラム作成・転送を再度実行して下さい。また、関数名、変数名、リスト名のいずれかに英数字以外の文字が使用されている場合も本メッセージが表示されます。関数名、変数名、リスト名には英数字のみを使用して下さい。



【未接続ブロックの例】



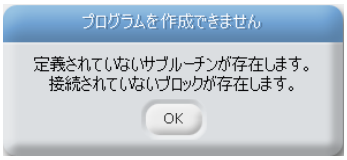
本メッセージは、関数ブロックが定義されていない場合



【未定義関数ブロックの例】

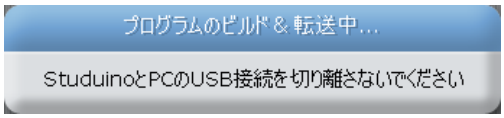
に表示されます。下図の例のように、sub 関数をコールするブロックを使用した場合、sub 関数を定義しなければなりません。ビルドを通す場合は、関数ブロック

をスクリプトエリアにドロップし定義するか、メイン関数に繋がれている「sub 関数をコールする」ブロックを削除して、プログラム作成・転送を再度実行して下さい。

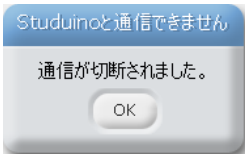


本メッセージは、上記 2 つのビルドエラーが同時に発生した場合に表示されます。ビルドを通す場合は、関数ブロックに繋がれていないブロックに対応し、「関数をコールする」ブロックでコールしている関数ブロックに対処して、プログラム作成・転送を再度実行して下さい。

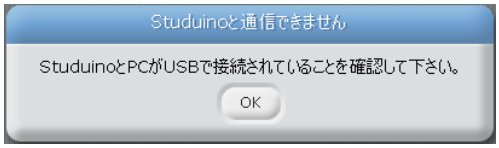
ビルドに成功すると、作成したプログラムを Studuino に転送します。下記に転送時に表示されるメッセージを記します。



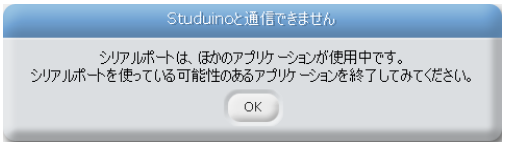
本メッセージは、作成したプログラムを Studuino に転送している際に表示されます。本メッセージが表示されている間は、Studuino と PC の USB 接続を切り離さないでください。もし、本メッセージ表示中に USB 接続が切り離された場合、下記メッセージが表示されます。



本メッセージは、プログラム転送中に Studuino と PC の USB 接続が切り離された場合に表示されます。Studuino と PC を再度 USB 接続し、プログラム作成・転送を実行して下さい。



本メッセージは、Studuino と PC が USB 接続されていない場合に表示されます。本メッセージが表示された場合、Studuino と PC が USB 接続していることを確認し、再度プログラム作成・転送を実行して下さい。



本メッセージは、Studuino と PC を接続している通信が他のアプリケーションによっ

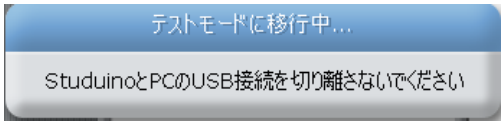
て使用されている場合に表示されます。Studuino と接続している可能性のあるアプリケーションを終了し、再度プログラム作成・転送を実行して下さい。

- **プログラム実行**

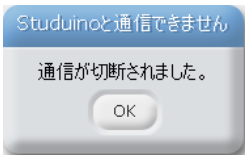
  Studuino に転送したプログラムを実行します。メニュー項目「プログラム作成・転送」成功後に、表示されます。

- **テストモード開始/終了**

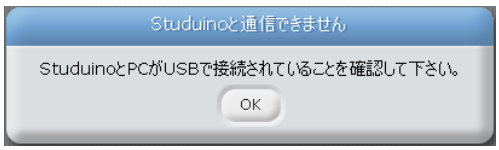
  Studuino と通信し、Studuino に接続されているパーツをリアルタイムに制御します。テストモード時と通常時でメニュー項目(「テストモード開始」/「テストモード終了」)が切り替わります。テストモードを開始する前に必ず Studuino と PC が USB 接続されていることを確認して下さい。



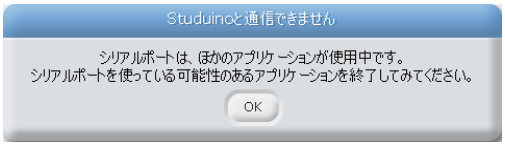
テストモードを開始すると、左図のメッセージボックスを表示し、テストモードに移行します。テストモードを開始してからテストモードを終了するまでは、Studuino と PC の USB 接続を切り離さないでください。もし、本メッセージ表示中に USB 接続が切り離された場合、下記に示すメッセージボックスが表示されるか、タイミングによっては、Studuino ブロックプログラミング環境がフリーズする場合がありますので、テストモードを開始したら終了するまでは絶対に Studuino と PC の USB 接続を切り離さないで下さい。下記にエラー時に表示されるメッセージボックスとエラーの内容を記します。



本メッセージは、テストモード移行中に Studuino と PC の USB 接続が切り離された場合に表示されます。Studuino と PC を再度 USB 接続し、テストモード開始を実行して下さい。

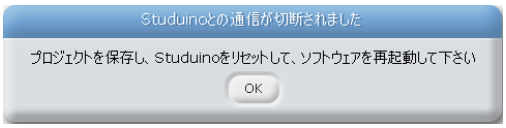


本メッセージは、Studuino と PC が USB 接続されていない場合に表示されます。本メッセージが表示された場合、Studuino と PC が USB 接続されていることを確認し、再度テストモード開始を実行して下さい。



本メッセージは、Studuino と PC を接続している通信が他のアプリケーションによって使用されている場合に表示されます。Studuino と接続している可能性のあるアプリケーションを終了し、再度テストモード

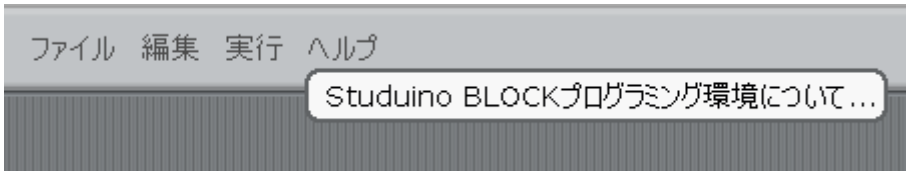
開始を実行して下さい。



本メッセージは、テストモード時に PC の USB 接続が切り離された場合に表示されます。作成したプログラムを保存し、Studuino ブロックプログラミング環境を再起動して下さい。

■ ヘルプメニュー

ヘルプメニューでは、Studuino プログラミング環境についての情報を提供します。

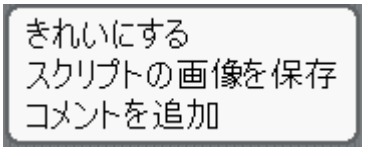


- **Studuino ブロックプログラミング環境について...**

  Studuino ブロックプログラミング環境の情報を示します。

5.6. コンテキストメニュー

スクリプトエリア上で右クリックするとスクリプトに置かれているブロックに対するコンテキストメニューが表示されます。



- **きれいにする**

  スクリプトエリアに置かれたブロックを整理します。

- **スクリプト画像を保存**

  スクリプトエリアに置かれたブロックを gif 形式で保存します。

- **コメント追加**

  コメントを追加します。表示されたコメントを処理ブロック上にドラッグ＆ドロップすることで、ドロップされた処理ブロックとコメントを対応付けることができます。

スタートブロック

以外のブロック上で右クリックするとクリックしたブロックに対するコンテキストメニューが表示されます。

- **複製**

  ブロックを複製します。

- **削除**

  ブロックを削除します。

6. トラブルシューティング

本ソフトウェアを使用した際のトラブルシューティングを下記に記します。発生した問題が以下に記載した手順で解決できない場合や発生した問題が記載されていない場合は、お手数ですが末尾のご連絡先までお問い合わせください。また、公式サポートページも立ち上げ、随時サポート情報を更新することを予定しておりますので、こちらもご確認ください。

http://www.artec-kk.co.jp/studuino

6.1. Studuino プログラミング環境共通

Studuino アイコンプログラミング環境／Sturuino ブロックプログラミング環境共通のトラブルシューティングを下記に記します。

- **テストモードでサーボモーター、DC モーターが動かない**

サーボモーターと DC モーターは、電池 BOX からの電源供給により正常に動作します。電池 BOX にアルカリ電池が入っていること、電池 BOX が Studuino と接続されていること、電池 BOX の電源が ON になっていることを確認して下さい。

- **意図しないタイミングでロボットにリセットがかかる**

サーボモーターを使用している場合、アルカリ電池を使用しないと電力不足で Studuino にリセットがかかります。また、アルカリ電池が消耗している場合でもリセットがかかります。新しいアルカリ電池に交換して下さい。また、電源投入時は信号が不安定なため、数回リセットがかかる場合があります。

- **センサーの値が仕様範囲にならない**

センサーは個体差があり、表示される値に多少のばらつきがあります。

6.2. Studuino アイコンプログラミング環境

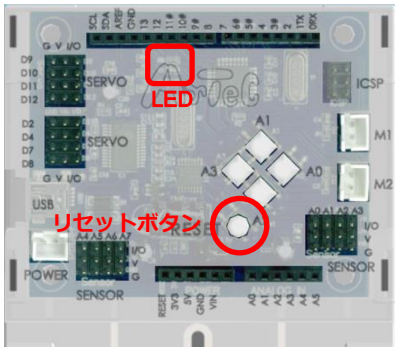
- **プログラム作成・転送時にエラーメッセージが表示される**
- **テストモードにならない**
- **センサー値確認モードにならない**

下記が表示された場合は、Studuino が停止している可能性があります。

タイトル	メッセージ
Studuino と通信できません	Studuino と PC の同期が取れませんでした。このプロジェクトを保存し、プログラミング環境を終了し、Studuino をリセットして、プログラミング環境を再度起動して下さい。

プログラムを保存し、プログラミング環境を再起動した後、下記の手順で Studuino のリセットを行ってください。

1．電池ボックスの電源を切り、Studuino に接続している USB ケーブルを抜く
2．３秒ほど待つ
3．Studuino に USB ケーブルを接続し、電池ボックスの電源を入れる
4．Studuino のリセットボタンを押す。この時に、LED が緑色に点滅することを確認して下さい(右図参照)



上の手順を行った後、テストモードを実行して下さい。

下記が表示された場合は、インストールした Studuino プログラミング環境のシステムファイルが壊れている可能性があります。

タイトル	メッセージ
システムファイルが壊れています	Studuino プログラミング環境を再インストールして下さい。

本メッセージが表示された場合、強制的に Studuino アイコンプログラミング環境を終了しますので、表示されるファイル保存ダイアログで作成したプログラムを保存して、別紙 **Studuino ソフトウェアのインストール**の手順に従って、Studuino プログラミング環境をアンインストールした後、再インストールして下さい。

タイトル	メッセージ
Studuino と通信できません	COM ポートでエラーが発生しています

本メッセージは、PC 側で Studuino との USB 接続を開始できない場合に表示されます。PC を再起動することで必要があります。

上記以外のメッセージが表示された場合は、**4.5.メインメニュー**の【プログラム作成・転送】、【テストモード】または、【センサー値確認モード】に原因と対策を記載していますので、そちらを参照して下さい。

6.3. Studuino ブロックプログラミング環境

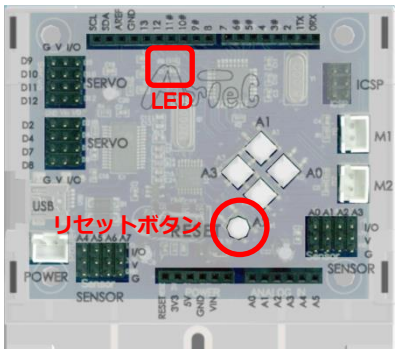
- **テストモード実行時にエラーメッセージが表示される。**
- **プログラム作成・転送時にエラーメッセージが表示される。**

下記が表示された場合は、Studuino が停止している可能性があります。

タイトル	メッセージ
Studuino と通信できません	Studuino と PC の同期が取れませんでした。このプロジェクトを保存し、プログラミング環境を終了し、Studuino をリセットして、プログラミング環境を再度起動して下さい。

プログラムを保存し、プログラミング環境を再起動した後、下記の手順で Studuino のリセットを行ってください。

1．電池ボックスの電源を切り、Studuino に接続している USB ケーブルを抜く
2．３秒ほど待つ
3．Studuino に USB ケーブルを接続し、電池ボックスの電源を入れる
4．Studuino のリセットボタンを押す。この時に、LED が緑色に点滅することを確認して下さい(右図参照)



上の手順を行った後、テストモードを実行して下さい。

下記が表示された場合は、インストールした Studuino プログラミング環境のシステムファイルが壊れている可能性があります。

タイトル	メッセージ
システムエラーが発生しました	システムファイルが壊れています。このプロジェクトを保存し、プログラミング環境を終了し、Studuino プログラミング環境をアンインストールして再インストールして下さい。

本メッセージが表示された場合、作成したプログラムを保存して、Studuino ブロックプログラミング環境を終了し、別紙 **Studuino ソフトウェアのインストール**の手順に従って、Studuino プログラミング環境をアンインストールした後、再インストールして下さい。

タイトル	メッセージ
Studuino と通信できません	COM ポートでエラーが発生しています

本メッセージは、PC 側で Studuino との USB 接続を開始できない場合に表示されます。PC を再起動することで必要があります。

上記以外のメッセージが表示された場合は、**5.5** メインメニューの【プログラム作成転送】または、【テストモード開始/終了】に原因と対策を記載していますので、そちらを参照して下さい。

- **プログラム作成・転送時にエラーメッセージが表示されます。(プログラム作成時)**

プログラム作成時に下記が表示された場合は、インストールした Studuino プログラミング環境のシステムファイルが壊れている可能性があります。

タイトル	メッセージ
プログラムを作成できません	システムエラー1が発生しました。再インストールして下さい。
	システムエラー2が発生しました。再インストールして下さい。
	システムエラー3が発生しました。再インストールして下さい。

本メッセージが表示された場合、作成したプログラムを保存して、Studuino ブロックプログラミング環境を終了し、別紙 **Studuino ソフトウェアのインストール**の手順に従って、Studuino プログラミング環境をアンインストールした後、再インストールして下さい。

- **プログラム作成時にプログラムがオーバーフローする**

Studuino ブロックプログラミング環境で作成できるプログラムの大きさは、Studuino の性能に依存します。下記のメッセージは、作成したプログラムが Studuino のプログラムサイズ約 15Kbyte を超えた時に表示されます。

タイトル	メッセージ
プログラムを作成できません	スクリプトが大きすぎます。制御ブロック等を利用してスクリプトを小さくして下さい。

本メッセージが表示された場合、制御ブロック等を利用してプログラムサイズを小さくしてください。また、Studuino ブロックプログラミング環境は、大規模プログラム開発を対象としていません。比較的小規模なプログラミングを対象とした設計となっていますので、もし大規模プログラム開発を希望の場合は、Arduino IDE や Atmel Studio を使用する事を検討して下さい。

7. お問い合わせ先

株式会社 ARTEC お客様相談窓口

お電話によるお問い合わせ 072-990-5656

E メールによるお問い合わせ info@artec-kk.co.jp